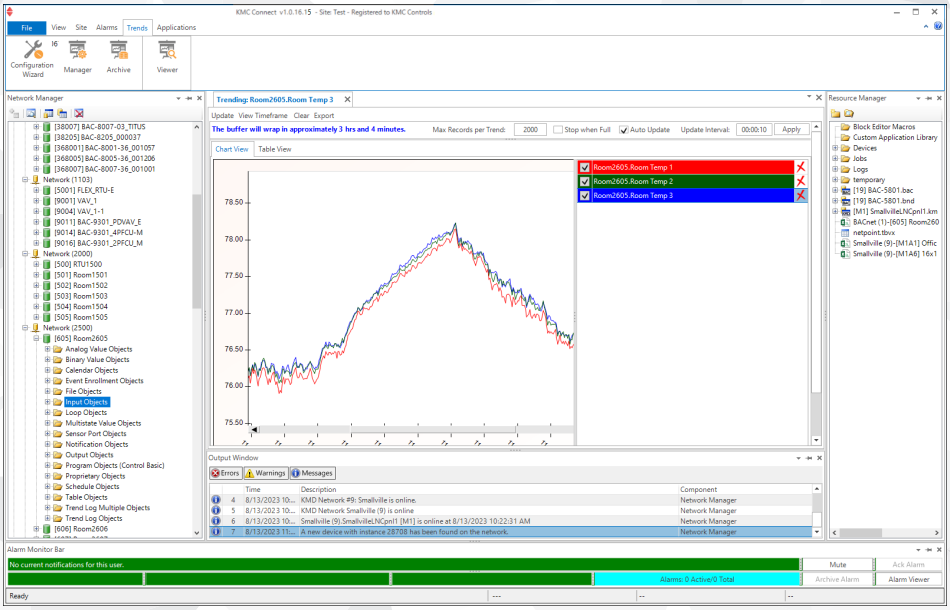
KMC Connect™
Hardware Configuration Software

KMC Connect™ Reference

Includes installation instructions



KMC CONTROLS

Revision N

©2025, KMC Controls, Inc.

NetSensor, WinControl, and the KMC logo are registered trademarks of KMC Controls, Inc.

AppStat, BACstage, FlexStat, FullBAC, KMC Connect, TotalControl, SimplyVAV, and the SimplyVAV logo are trademarks of KMC Controls, Inc.

Patent information available at https://www.kmccontrols.com/patents/.

ActiveX, Silverlight, Internet Explorer, and Microsoft Excel, Windows, and Windows Vista are registered trademarks of Microsoft, Inc.

BACnet and ASHRAE are registered trademarks of the American Society of Heating, Refrigerating and Air-Conditioning Engineers.

Adobe Flash Player, Acrobat and Acrobat Reader are trademarks of Adobe Systems, Inc.

Cimetrics and BACstac are trademarks of Cimetrics, Inc.

Java is a registered trademark of Oracle, Inc.

Firefox is a registered trademark of the Mozilla Foundation.

Google Chrome is a registered trademark of Google LLC.

All rights reserved. No part of this publication may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language in any form by any means without the written permission of KMC Controls, Inc.

Printed in U.S.A.

The material in this manual is for information purposes only. The contents and the product they describe are subject to change without notice. KMC Controls, Inc. makes no representations or warranties with respect to this manual. In no event shall KMC Controls, Inc. be liable for any damages, direct or incidental, arising out of or related to the use of this manual.

KMC Controls, Inc.
19476 Industrial Drive
New Paris, IN 46553
U.S.A.
TEL: 1.574.831.5250
FAX: 1.574.831.5252
info@kmccontrols.com

Contents 3

About this manual 13
What you need to know 13
Conventions used 13
If you encounter difficulty 14

Part I: Setting up KMC Connect 15

Section 1: About KMC Connect 17
Computer requirements 18
Setting up computers 20
Computer names 20
Passwords and privileges 20
Sleep and hibernation modes 20
Installing KMC Connect 20
Uninstalling KMC Connect 21

Section 2: Getting started 23
Starting KMC Connect 24
Using a Remote License Key 25
Adding a new site 27
Open an existing site 29
Closing a site 30
Working offline 30

Part II: Working with KMC Connect 31

Section 3: The KMC Connect workspace 33
Customizing the workspace 35
Options for KMC Connect 38
Object Locate 39

Section 4: Using the Network Manager 41
Opening and closing the Network Manager pane 42
Network Manager toolbar commands 42
Discovering devices 43
Setting discovery options 43
Adding a service 44
Configuring an existing service 46
Removing a service 47
Changing the Network Manager options 48
Naming BACnet networks 49
Renaming BACnet objects in the Network Manager 49
Opening KMC Conquest device served web pages 50
Restarting BACstac from the Network Manager 51
Enable and disable NFC 52

Section 5: The Output Window 53

Opening and closing the Output Window 54
Sorting and clearing the Output Window 54
Output Window options 54

Section 6: About the Resource Manager 57
Opening and closing the Resource Manager pane 58
Editing items in the Resource Manager list 58
Restoring a configuration 59
Creating backup files 62
Managing the Resource Manager list 62
Adding files to the Resource Manager folder 63
Renaming BACnet objects in the Resource Manager 64

Section 7: Viewing objects with Table View 65
Viewing groups of similar objects 66
Custom Table Views 69

Section 8: Backing up and restoring a site 73
Backing up a site 73
Restoring site data 74

Section 9: Security and passwords 77
KMC Connect Basic Security 77
KMC Connect advanced security 80
Group roles for security 81
Adding and modifying groups 83
Adding TotalControl users to a KMC Connect site 86

Section 10: The KMC Connect Audit Log 91

Section 11: Upgrading firmware 93

Part III: Trend logs 97

Section 12: About trend logs 99

Section 13: Viewing trend logs in KMC Connect 101
Viewing with the Trend Viewer 102
Starting a live trend log 102
Starting from the Network Manager 103
Viewing trend logs in the Trend Manager 103
Parts of the Trend Viewer 104
Viewing controller trend logs 106
Archiving trend logs 107

Section 14: Configuring BACnet trend logs and groups 109
Trend Configuration Wizard 110
Start the wizard 110
Add objects by searching 112
Configuring BACnet controller trend logs 113
Configuring BACnet device trend logs 114
Configuring a BACnet PC trend log 117
Adding BACnet trend logs to trend groups 119

Section 15: Configuring KMD trend logs and groups 121
Configuring KMD controller Trend Logs 121
Configuring KMD device trend logs 123
Configuring KMD PC trend logs 125
Adding KMD trend logs to a trend group 126
Section 16: Configuring the Trend Service 129

Part IV: Alarms, notifications, and events 131

Section 17: An overview of notifications, alarms, and events 133
Section 18: Configuring BACnet controllers for alarms 135
Using the Alarm Wizard for BACnet alarms 136
Start the wizard 136
Add by searching 138
About intrinsic BACnet alarms and events 139
An example of intrinsic notification 140
Notifications in analog objects 140
Notifications in binary objects 142
Configuring BACnet devices for intrinsic alarms 143
About algorithmic reporting in BACnet devices 144
Configuring event enrollment objects 146
Configuring a notification class object 148
Section 19: Configuring KMD controllers for alarms 151
Control Basic alarm statements 151
ALARM statement 151
DALARM statement 152
Alarm classes for KMD controllers 153
Setting up KMD alarms 154
Section 20: Change Of Value (COV) notifications 155
The parts of COV 155
COV server 155
COV client 156
Subscribing to an object on a configuration page 157
Subscribing to a property on a configuration page 158
Unsubscribing from COV subscriptions 159
Subscribing trend logs to a COV server device 160
Viewing all event enrollment subscriptions 160
Section 21: Viewing, acknowledging, and archiving alarms and events 163
Viewing alarms 163
Acknowledging alarms 165
Archiving alarms in KMC Connect 166
Viewing a summary of BACnet notifications 166
Using the Alarm Monitor bar 167
Filtering alarms 168
Disabling alarms by network 169
Section 22: Emailing operator alarms 171

Setting up alarm emails 171
The components of the Alarm Routing Manager 171
Changing the listing view 172
Configuring email notifications 172
Setting up emails for alarms and events 173
The components of the Alarm Routing Manager 174
Changing the listing view 174
Section 23: Filtering alarms by security groups 175
Section 24: Configuring the Notification Service 177

Part V: Schedule management 181

Section 25: About BACnet schedules 183
Configuring BACnet Schedule objects 184
Entering time-value pairs 184
Configuring an Exception schedule 185
Configuring BACnet Calendar objects 186
Section 26: About KMD Schedules 189
Configuring KMD Weekly Schedule points 190
Configuring KMD Annual Schedule points 191

Part VI: Control Basic and the Code Editor 193

Section 27: Programming with the Code Editor 195
Writing Control Basic programs in controllers 196
Writing programs offline 198
Automatically numbering program lines 200
Keyword prompting 200
Using keyboard shortcuts 201
Finding and replacing text 202
Changing Code Editor options 206
Reading live values 210
Locking Control Basic programs 211
Adding objects by dragging 214
Section 28: About Control Basic programs 217
Line vs. block programming 218
About Control Basic scans 218
Writing Control Basic statements 219
Statements 219
Multiple statements 219
Functions 219
Expressions 220
Labels and line numbers 220
Line numbers–Standard Control Basic only 221
Labels in Next Generation Control Basic 221
Programming format and notation 222
Real numbers 222

Hierarchy of operators 223
Relational operators 224
Using arithmetic operators 225
Using Boolean logic 225
Programming with variables 226
Variables in KMD controllers 226
BACnet value objects as variables 226
Local variables 226
Reading and writing BACnet objects 227
Mnemonics for BACnet controllers 227
Name/Mnemonic switching 229
Reading and writing the priority array 231
Transferring values between BACnet controllers 232
Limits on transferring off-panel BACnet properties 232
Using WAIT statements to transfer values 233
Using NETPOINT and NETPOINTCACHE to transfer values 234
Reading and writing off-panel points with COV 235
Using WAIT statements to transfer values 237
Using INTERVAL for reading off-panel points 237
Reading and writing KMD points 239
Mnemonics for KMD controllers 239
Transferring values with KMD want-points 240
KMD want-point limitations 241
User-defined functions and procedures 244
BACnet arrays and dynamic access 246
BACnet data types 247
Generation 5 data types 248

Section 29: Keywords for Control Basic 251
Using example programs from help 251
Syntax for commands and functions 251

Section 30: Writing block programs 339
Applicable controllers 339
Starting the Block Editor 341
A tour of the Block Editor 343
The parts of the editor 343
The Block Editor toolbar 344
Get and Set blocks 344
Get blocks 344
Set blocks 345
Defining and using local variables 345
Adding and editing routines 346
Macro blocks 347
Limitations and importing line programs 350

Section 31: Reference to Control Basic blocks 353

Part VII: BACnet objects 399

Section 32: BACnet overview 401
BACnet objects and properties 401
An example BACnet object 401
BACnet properties 401
BACnet services 402
Priority arrays 402
An example priority array 403
Standard BACnet priority levels 405

Section 33: Configuring and connecting to BACnet devices and networks 407
Adding BACnet devices to the Network Manager list 408
Configuring BACnet devices and objects 410
Converting input and output objects 412
Configuring a BACnet service 413
Regenerating networks and devices 415
Reinitializing a BACnet device 416
Setting BACnet system time 417
Automatic MAC Addressing commands 418
Scanning BACnet configuration files 419
Creating and deleting BACnet objects 420
Configuring the IP connection 421
Comparing BACnet objects 423

Section 34: Backing up and restoring BACnet and KMDigital devices 425
Backing up with .bnd files 425
BACnet backup and restore 426
Importing and exporting BACstage files 428
Scheduling BACnet and KMDigital devices for backup 429

Section 35: Reference to BACnet objects 435
Access Door objects 435
Averaging objects 440
Calendar objects 441
Command object 442
Device objects 443
Display objects 450
Event enrollment objects 451
Event Log objects 455
File objects 457
Group objects 457
Input objects–accumulator 458
Input objects–analog 461
Input objects–binary 466
Input objects–multistate 469
Life Safety Point objects 472
Life Safety Zone objects 476
Load Control objects 480
Loop objects 484
Notification objects 490

Output objects–analog 491
Output objects–binary 494
Output objects–multistate 498
Program objects 501
Schedule objects 504
Structured View objects 508
Trend Log objects 509
Trend Log Multiple objects 514
Value objects–analog 518
Value objects–binary 521
Value objects–multistate 525

Section 36: Configuring Sensor Port objects 529
Configuring KMD series NetSensors for BACnet 529
Configuring the STE-9000 series NetSensor 535
Configuring the HPO-9007 wireless gateway 541

Section 37: Configuring Conquest FlexStats 551
Conquest FlexStat Configuration Tabs 552

Section 38: BACnet tables 561
Input tables 561
CB tables 562

Section 39: Supported engineering units 565

Section 40: Reference to KMC Controls BACnet controllers 567

Part VIII: KMDigital networks and points 571

Section 41: Configuring and connecting to KMDigital controllers and networks 573
Before connecting to a KMDigital network 573
Adding a KMDigital Tier 1 network 574
Adding a Tier 1 network connection 574
Adding a Tier 1 serial connection 576
Adding a KMDigital Tier 2 network 579
Configuring KMD controllers and points 581
Updating the Network Manager list 583
Configuring a KMD service 584
Configuring a KMD network 586
Setting time for KMD networks 587

Section 42: Reference to KMD points 589
Annual Schedule points 589
Array points 591
Device points 592
Input points-analog 594
Input points-binary 595
InterProtocol points 596
Configuring a NetSensor for KMDigital controllers 599
Output points-analog 605
Output points-binary 607

PID loop controllers 608
Control Basic programs 611
Runtime Log points 611
System Groups 613
Tables 614
Trend Log points 615
Variable points-analog 616
Variable points-binary 617
Weekly Schedule points 618

Section 43: Supported KMD controller engineering units 621

Section 44: Reference to KMD controllers 623

Section 45: KMD Tier 1 alarm messages 625

Part IX: Working with OPC objects 629

Section 46: An introduction to OPC 631

Section 47: Setting up an OPC service 633
Connecting to an OPC server 633
Configuring an OPC service 634

Section 48: Reference to OPC objects and tags 637

Part X: Protocol Gateway Service and Protocol Gateway Manager 641

Section 49: Overview of the Protocol Gateway 643

Section 50: Setting up Protocol Gateway processes 645
Using the Protocol Gateway Manager 645
Managing Protocol Gateway processes 647

Section 51: Configuring the Protocol Gateway service 651

Part XI: Applications and wizards 653

Section 52: Application selection, configuration, and deployment 655
Step 1—Selection 655
Step 2—Configuration 656
Step 3—Deployment 657

Section 53: Audit application 659

Section 54: VAV Balancing application 663
The VAV Balancing sequence 663
VAV flow monitoring 666
Commanding VAV airflow 666

Section 55: Custom Applications Wizard 669

Section 56: Input and output object wizards 673
Analog input object wizard 673
Binary input object wizard 676

Analog output object wizard 677

Appendices 679

Appendix A: Communication ports and BAS networks 681
Firewalls and ports 681
Remote access 683
BACnet controllers and networks 684
BACnet networks, broadcast domains and tunneling 684
Bandwidth and traffic loading 685
Other applications and KMC Connect on a single computer 685
KMD controllers and networks 685
Bandwidth issues 686
Summary of KMD network use 686
Network protocols supported 687
Broadcasting on KMD networks 687

Appendix B: Configuring the Cimetrics BACstac driver 689
Before you begin 689
Configuring BACstac for BACnet IP 690
Configure the computer as a BACnet IP device 690
Register as a foreign device 692
Configure BACstac as a BBMD 693
Configure BACstac for Ethernet 8802.3 694
Remove previous versions of the BACstac driver 695

Appendix C: Versions of Control Basic 697
Control Basic versions in controllers 697
Deprecated keywords 698
Changes to IF THEN 700
File compatibility 700
TotalControl and KMC Connect compatibility 700
BACstage file compatibility 700
New keywords 701
Next Generation 701
Generation 5 702
Line numbers and labels 705
Line numbers 705
Labels 705
Local variables 706
References to objects in remote devices 706
References to properties 706

Appendix D: KMC Conquest-compliant controllers 709

Appendix E: Glossary 711

Index 717

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 managed with KMC Connect.

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, Inc. provides the following assistance.

Printed version of help An Adobe Acrobat version of KMC Connect help is available from the KMC Controls website. The PDF document is identical to the online 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 Time 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 and KMDigital 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 available from the Software downloads area in the Partner Portal on the 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

- *Computer requirements* on page 18
- *Setting up computers* on page 20
- *Installing KMC Connect* on page 20
- *Uninstalling KMC Connect* on page 21

Computer requirements

Computers running KMC Connect must meet the minimum hardware and operating system requirements listed in the tables in this topic. Review these requirements before installing KMC Connect.

KMC Connect will run on either 32-bit or 64-bit versions of Windows. It can be installed on hot-swappable or replicated servers as long as the servers meet the minimum operating system requirements for KMC Connect.

Table 1–1 Windows 11 computer requirements

Component	Requirements
Operating system	Windows 11 Professional
Processor	Two or more cores, 32- or 64-bit
Trusted Platform Module (TPM)	Yes
Processor speed	2 GHz or faster
RAM	8 gigabyte (GB) minimum, 16 GB recommended
Graphics adapter	DirectX 9 or greater graphics processor
Monitor resolution	1920 x 768 pixels or higher
Network connection	10BaseT Ethernet miminum, 100BaseT Ethernet recommended
Hard drive	500 GB minimum, 1 terabyte (TB) recommended
USB port	1 dedicated to Hardware License Key

Table 1–2 Windows 10 computer requirements

Component	Requirements
Operating system	Windows 10 Professional
Processor	Two or more cores, 32- or 64-bit
Trusted Platform Module (TPM)	No
Processor speed	2 GHz or faster
RAM	8 gigabyte (GB) for 32-bit or 64-bit
Graphics adapter	DirectX 9 or greater graphics processor
Monitor resolution	1920 x 768 pixels or higher
Network connection	10BaseT Ethernet miminum, 100BaseT Ethernet recommended
Hard drive	500 GB minimum, 1 terabyte (TB) recommended
USB port	1 dedicated to Hardware License Key

Table 1–3 Windows Server 2022 computer requirements

Component	Requirements
Operating system	Windows Server 2022
Processor	Two or more cores, 32- or 64-bit
Trusted Platform Module (TPM)	Yes
Processor speed	2 GHz or faster
RAM	16 GB or greater
Hard drive	1 TB

Table 1–4 Windows Server 2019

Component	Requirements
Operating system	Windows Server 2019
Processor speed	Two or more cores, 32- or 64-bit
Trusted Platform Module (TPM)	No
Processor speed	2 GHz or faster
RAM	16 GB or greater
Hard drive	1 TB

Setting up computers

Computers hosting KMC Connect must meet the requirements listed in the section *Computer requirements* on page 18. Before installing KMC Connect, verify the computer name, passwords and privileges, and 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. Use the Help function in Windows to search for “power” or “power management”.
2. Choose the Help topic for changing or setting up a power scheme.
3. In the power scheme, change the computer power settings to disable hibernation and set the sleep mode to “never”.

Installing KMC Connect

KMC Connect installation files are distributed in a `.zip` file available from the Downloads section of our web site at partners.kmccontrols.com. All of the files for KMC Connect are included in the `.zip` file.



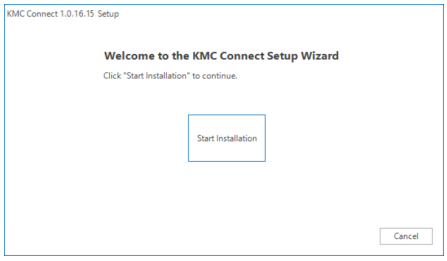
Note: A hardware license key is not required to install KMC Connect. However, you will need a key to start KMC Connect after installation.

To install KMC Connect, do the following.

1. In the Downloads folder, right-click the `.zip` folder containing KMC Connect.
2. Temporarily disable any security features that might block the installation. For example, in Windows: > Right-click the `.zip` folder. > In the shortcut menu, click **Properties**. > In the **Security** section, select **Unblock**. > Click **Apply**.
3. Right-click the **KMC Connect** `.zip` folder.
4. Select **Extract All**.

5 Open the extracted installation folder and double-click the **Setup** icon.

6 In the Welcome to the KMC Connect Setup Wizard page, click **Start Installation**.



7 When prompted, choose a location for the program and record this information. KMC Controls recommends the default location.

Once KMC Connect is installed, see the topic *Getting started* on page 23.

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 22 for a list of the programs.

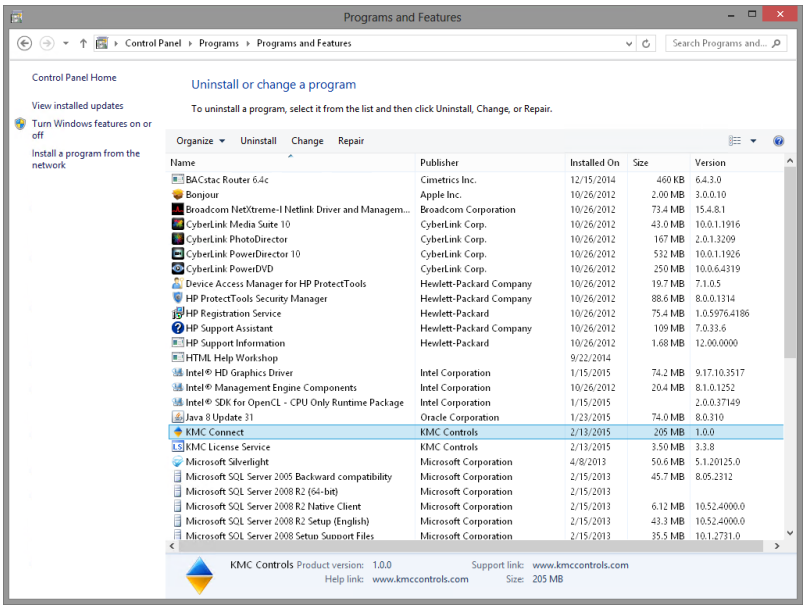
3 Choose **KMC Connect** and then click **Remove**.

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 and KMD 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 24
- *Adding a new site* on page 27
- *Open an existing site* on page 29
- *Closing a site* on page 30
- *Working offline* on page 30

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. For Windows 7, click **Start** > **Programs** > **KMC Controls** > **KMC Connect**. For Windows 10, click **Start** > **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

- *Using a Remote License Key* on page 25
- *Adding a new site* on page 27
- *Open an existing site* on page 29
- *Closing a site* on page 30
- *Working offline* on page 30

Using a Remote License Key

In addition to a hardware key license stored on a USB drive and a soft key license stored on the computer running KMC Connect, a remote license may be used to operate KMC Connect.

A common situation for a remote license connection is when using a virtual machine with no USB port for a hardware license key to host KMC Connect.

In order to use the remote license service, the remote server machine must have one of the following.

- A KMC License Service installed and running with a USB hardware license key ("hard key") inserted into the remote license server machine
- A hard key license converted to a soft key license stored on the computer running KMC Connect



Note: If the computer that stores the remote license is also running KMC Connect or TotalControl, some license features will be “in use” and unavailable through a remote license connection.



Note: The first time a hardware license key is plugged into a computer, Windows displays a message that new hardware has been found.

To use KMC Connect with a remote license, do the following:

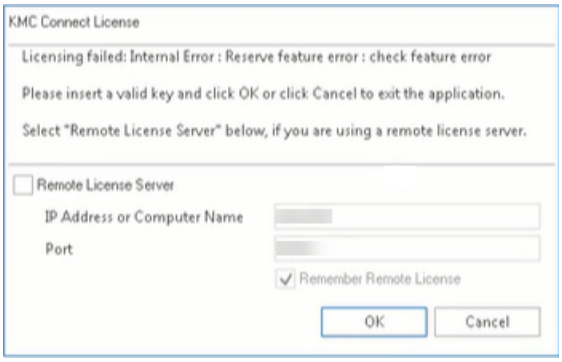
1. Verify that there is no hardware key inserted in a USB port on the computer on which KMC Connect is operating.
2. Start KMC Connect.

3 When the **Licensing failed...** dialog box opens, click to select the **Remote License Server** check box.

4 Type the IP address or computer name in the **IP Address or Computer Name** text box. To locate this information, click **File** > **Options** > **General Applications** > **Remote License**.

Note: The port number populates automatically.

Illustration 2–1 License dialog



5 For Windows 7, click **Start** > **Programs** > **KMC Controls** > **KMC Connect**. For Windows 10, click **Start** > **KMC Controls** > **KMC Connect**.

6 Once KMC Connect starts, you can choose a site to open. If a site has not been set up on the computer, it is possible to run KMC Connect without connecting to a site or setting up a new site.

Remember Remote License

To set up KMC Connect to apply the IP address or computer name automatically for each subsequent launch of the software using a remote license:

1 Click to select the **Remote License Server** check box.

2 Type the IP address or computer name in the **IP Address or Computer Name** text box. To locate this information, click **File** > **Options** > **General Application** > **Remote License**.

3 Select the **Remember Remote License** check box.

4 Click **OK**.

To delete the address and/or port information from the stored remote license:

1 Clear the information in the IP Address or Computer Name box and/or the Port text box.

2 Click **OK**.

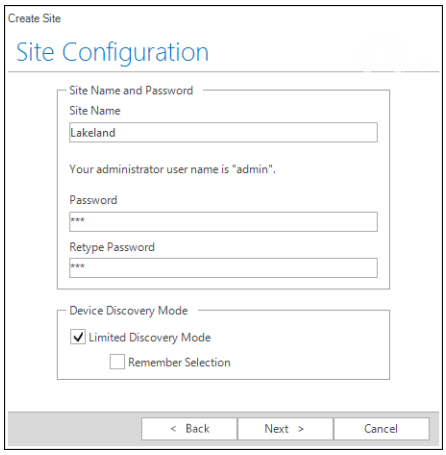
Adding a new site

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.

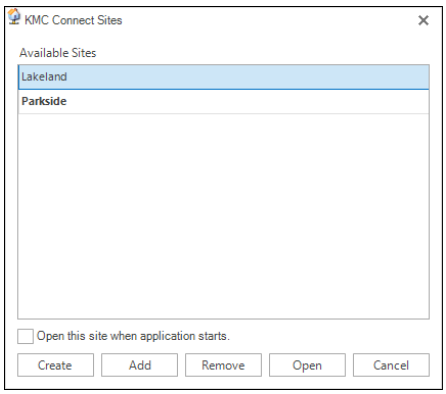
Note: The following procedure sets up a new site but does not configure BACstac for a connection to the BACnet internetwork. Verify or set Cimetrics BACstac to match the BACnet routing protocol, network, and port established by a BACnet router. See the topic *Configuring the Cimetrics BACstac driver* on page 689.

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

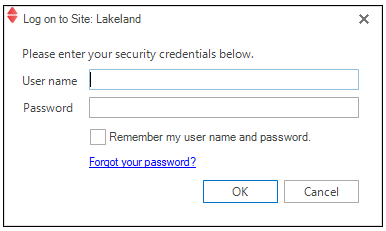
1. On the **Site** tab click **New**. The Site Name and Password dialog opens.
2. Enter the site name and a password for the site administrator.



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 413

Related topics

- *Open an existing site* on page 29
- *Closing a site* on page 30
- *Using a Remote License Key* on page 25
- *Working offline* on page 30

Open an existing site

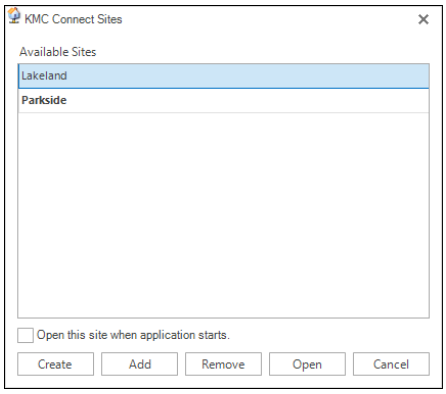
To open an existing site, you will need the following items.

- A hardware license key or a remote license key
- The user name and password established for the site

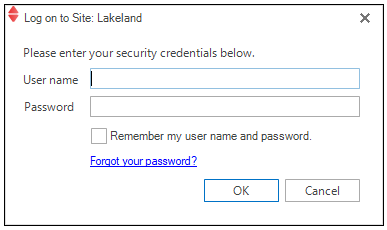
Note: The following procedure opens an existing site but does not configure BACstac for a connection to the BACnet internetwork. Verify or set Cimetrics BACstac to match the BACnet routing protocol, network, and port established by a BACnet router. See the topic *Configuring the Cimetrics BACstac driver* on page 689.

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 689.

Related topics

- *Adding a new site* on page 27
- *Closing a site* on page 30
- *Using a Remote License Key* on page 25
- *Working offline* on page 30

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 the Resource Manager* on page 57 to program and configure a device or object file in the Resource Manager.
- See *Application selection, configuration, and deployment* on page 655 to configure a job for Conquest compliant controller.

Related topics

- *Adding a new site* on page 27
- *Open an existing site* on page 29
- *Working offline* on page 30

Working offline

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

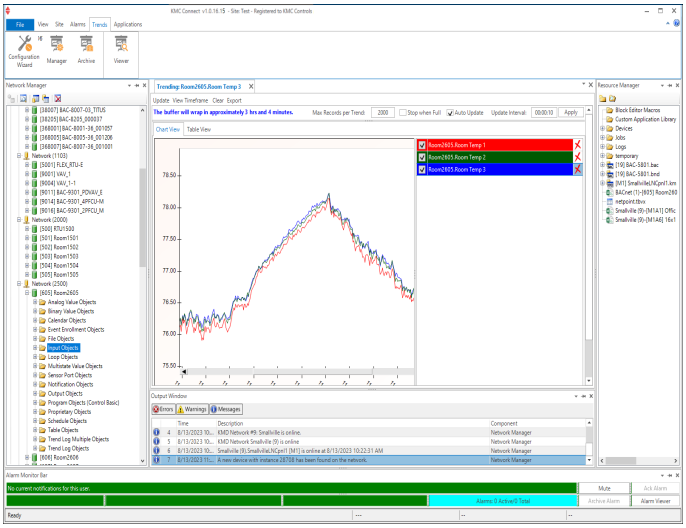
- See *About the Resource Manager* on page 57 to program and configure a device or object file in the Resource Manager.
- See *Application selection, configuration, and deployment* on page 655 to configure a job for Conquest compliant controllers.

Related topics

- *Adding a new site* on page 27
- *Open an existing site* on page 29
- *Closing a site* on page 30

KMC Connect

Part II: Working with KMC Connect

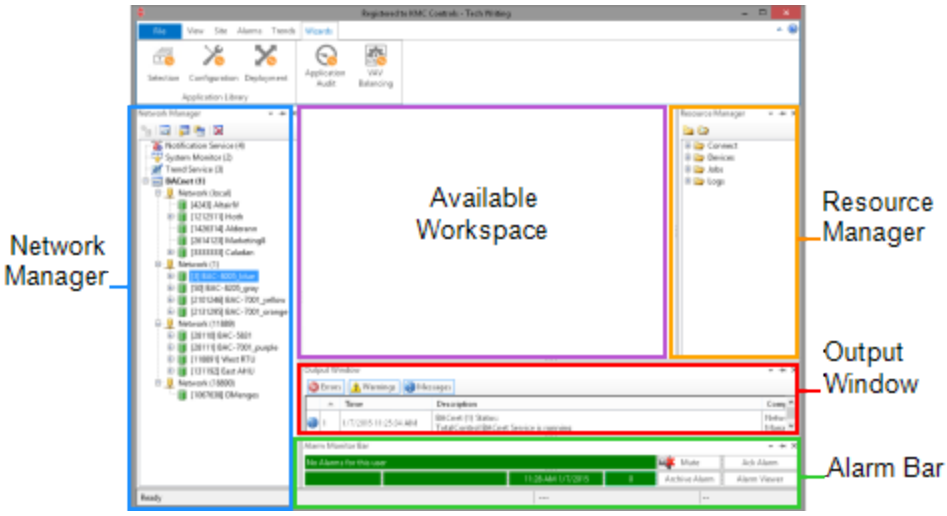


Section 3: The KMC Connect workspace

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

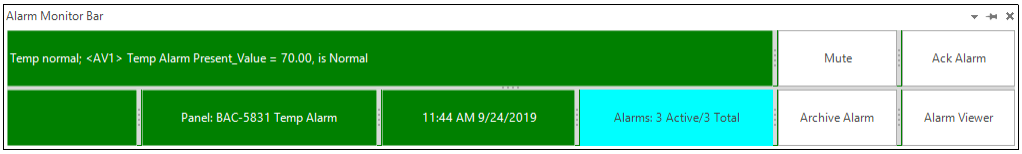


Site management tools The most frequently used components of KMC Connect are the dockable site management tools. Each tool opens in a separate pane in the KMC Connect workspace.

- The *The Output Window* on page 53 displays recent actions and messages about issues that may need operator attention.
- The Network Manager is an expandable list of all controllers, networks, and services in a KMC Connect-managed building automation system. See the topic *Using the Network Manager* on page 41 for all of the features of the Network Manager.
- The Resource Manager is a tool for managing templates and working offline to create controller configuration files, and Control Basic programs. See *About the Resource Manager* on page 57 for details on working offline.

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 167 for details on alarm management.

Illustration 3–2 Alarm Monitor Bar

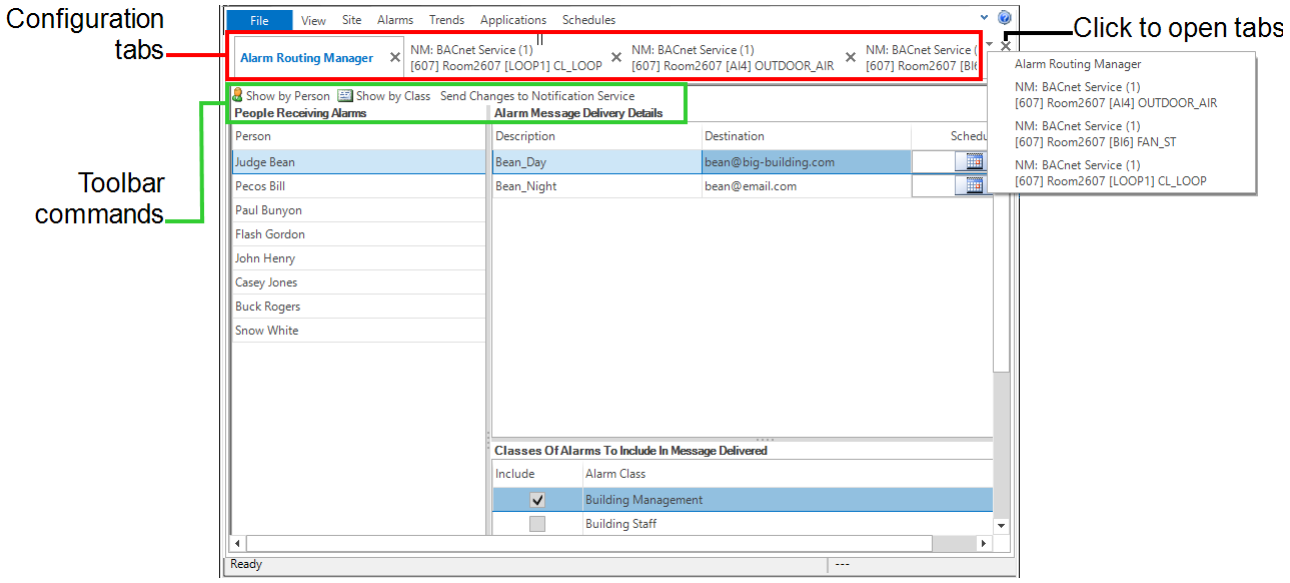


Configuration tabs Components in a KMC Connect-managed building automation site are set up and configured from configuration tabs. The tabs may be opened from the KMC Connect ribbon or from one of the site management tools.

- To configure a service, network, device or object, see *Using the Network Manager* on page 41.
- To set up and view data trends, see *About trend logs* on page 99.
- To write Control Basic programs, see *Control Basic and the Code Editor* on page 193.
- To set up alarms and notifications, see *Alarms, notifications, and events* on page 131.

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 3–3 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 downward 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**.

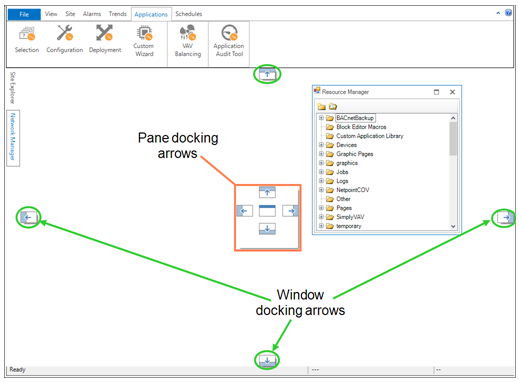
Related topics

- *Customizing the workspace* on page 35
- *Options for KMC Connect* on page 38
- *Object Locate* on page 39

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.

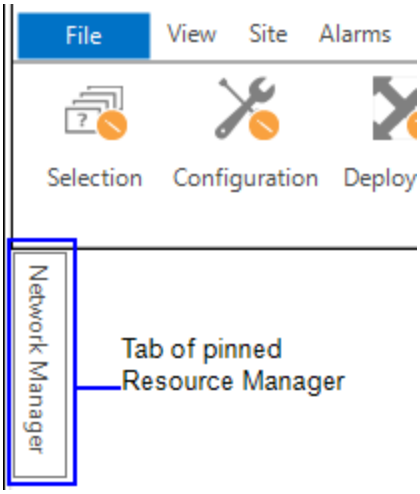
Illustration 3–4 Docking arrows



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 3–5 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 indicated 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:

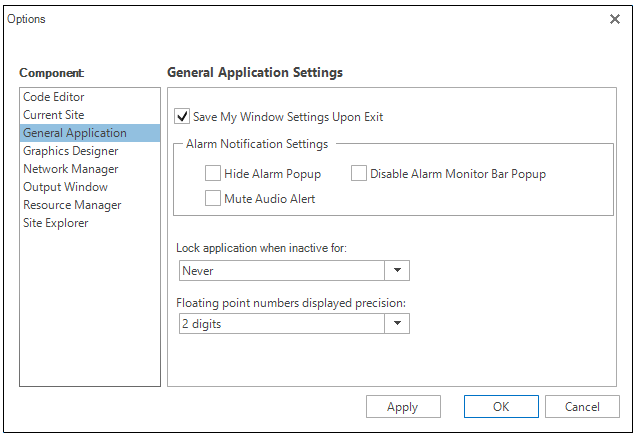
- On the KMC Connect toolbar, click the icon for the pane.
- From the **View** tab on the ribbon, choose the site management tool by name. Its pane will become active in the same location in which it was closed.

Options for KMC Connect

With the General Applications Options dialog, you can:

- Choose to save the position of the site management tools when KMC Connect closes.
- Control the way KMC Connect displays the Alarm Monitor Bar.
- Choose when KMC Connect starts an audible alarm after receiving an alarm or event notification.
- Set KMC Connect to lock after a period of inactivity.
- Set the display precision for floating point numbers.

Illustration 3–6 General Application options dialog



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

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

Save My Window Settings The Save My Windows Settings check box saves the configuration of the site management tools the next time KMC Connect opens.

- Select the **Save My Windows Settings Upon Exit** check box to set KMC Connect to save 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.

Alarm Notification Settings KMC Connect will open an alarm popup and sound an audible alarm when it receives a notification from a device on a building automation system network. Enable or disable the popup or audible alarm with the check boxes in this section of 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.
- Select the **Disable Alarm Monitor Bar Popup** check box to prevent the Alarm Monitor Bar from opening when a new alarm is received.

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. The default is two decimal places.

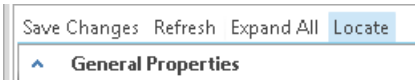
Object Locate

When working with sidebar panes, the sources of some files, objects and devices may become hidden due to scrolling or repositioning.

To bring the source of an open object, file or device back into focus in the Network Manager or Resource Manager, use the Object Locate feature.

Clicking **Locate** in the active item's toolbar will scroll to and highlight the item in its source pane.

Illustration 3–7 The Locate command in an open configuration tab



Related topics

- *Using the Network Manager* on page 41
- *About the Resource Manager* on page 57
- *Customizing the workspace* on page 35

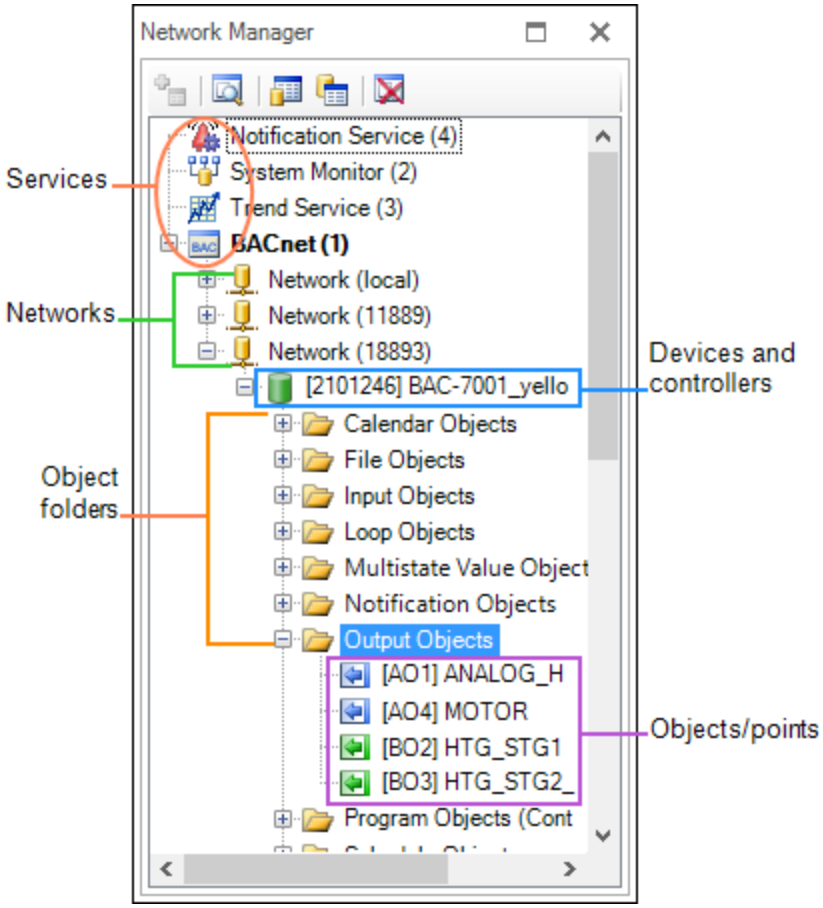
Section 4: Using the Network Manager

The 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 the 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 the Network Manager for the following:

- Display the structure of KMC Connect services, networks, and devices in the building automation system job site.
- As a source for dragging objects into Control Basic programs.
- 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 407
- *Configuring and connecting to KMDigital controllers and networks* on page 573
- *Setting up an OPC service* on page 633

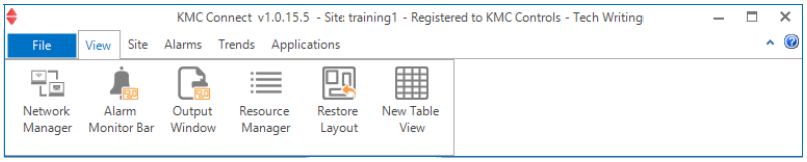
Related topics

- *Network Manager toolbar commands* on page 42
- *Opening and closing the Network Manager pane* on page 42
- *Changing the Network Manager options* on page 48

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 View tab on the ribbon

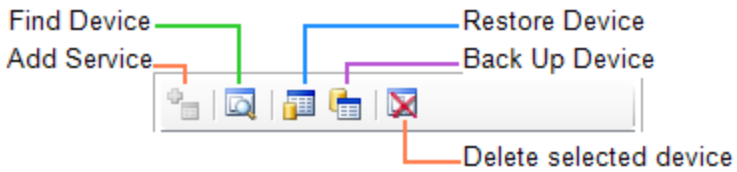


Network Manager toolbar commands

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 48.

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 Connect supports two 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 discovered under full discovery are represented by a full discovery icon .

To change the discovery method, see the topic *Setting discovery options* on page 43.

Initial discovery

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

Updating a network

1. In the Network Manager list, locate 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 list.

Complete discovery

When the site is set for Limited 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 408.

Setting discovery options

Discovery is the process of populating the Network Manager list with the devices on the network. KMC Connect supports two methods of discovery.

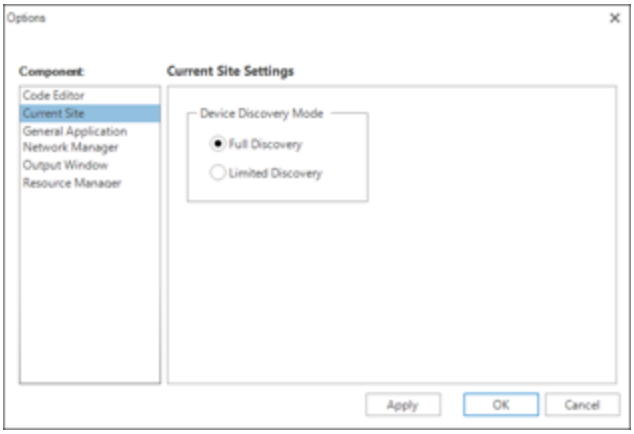
Limited discovery KMC Connect will read only the device object properties. This method is faster but may limit other functions. Devices are discovered by limited discovery are represented by the limited discovery icon .

Full discovery KMC Connect reads all objects and properties in every device. Devices discovered by full discovery are represented by the full discovery icon .

To change the discovery method, do the following:

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

Illustration 4–4 Current Site Settings dialog



See the related topic *Discovering devices* on page 43.

Adding a service

Adding a service connects one of the previously installed components of services to the site. Typically, the services are added and configured at the time they were installed. If a service was not connected to the site at the time of installation, add it with Network Manager.

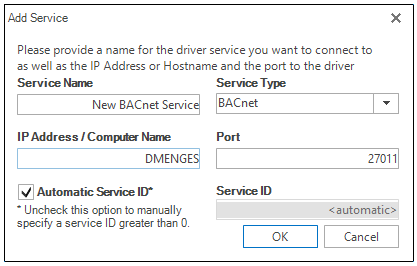
To add a service to the Network Manager list, do the following:

1. On the Network Manager toolbar, click the **Add Service** icon . The **Add Service** dialog opens.
2. Choose a service from the **Service Type** list.
3. Enter a descriptive name in the **Service Name** text box.
4. Enter either the IP address or the name of the computer hosting the SQL database.

5 If required, change the default number in **Port**.

6 When complete, click **OK**. The icon for the service is added to the Network Manager list.

Illustration 4–5 Add Service dialog



Details of the Add Service dialog

Service Name A descriptive name to display in the Network Manager list.

Service Type Choose a service to add from the **Service Type** list.

IP Address/Computer Name Enter the name or IP address of the computer hosting the site database.

Port Change the default port number only if required. For a listing of UDP ports used by KMC Connect, see *Communication ports and BAS networks* on page 681.

Automatic Service ID Select the **Automatic Service ID** check box when adding a new service. Clear the check box to change the Service ID number that was automatically assigned by KMC Connect.

Service ID Use only to change the Service ID number that was automatically assigned by KMC Connect.

Related topics

- *Using the Network Manager* on page 41
- *Network Manager toolbar commands* on page 42
- *Opening and closing the Network Manager pane* on page 42
- *Configuring the Trend Service* on page 129
- *Configuring the Notification Service* on page 177
- *Configuring and connecting to BACnet devices and networks* on page 407
- *Configuring and connecting to KMDigital controllers and networks* on page 573
- *Configuring the Protocol Gateway service* on page 651
- *Configuring an existing service* on page 46
- *Removing a service* on page 47

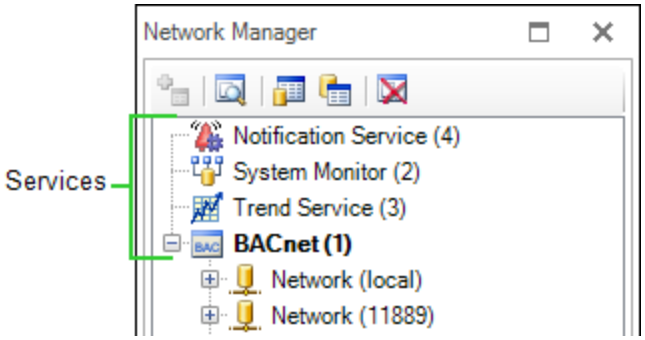
Configuring an existing service

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 the 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** in the configuration tab.

Illustration 4–6 Services in the Network Manager list



Related topics

- *Using the Network Manager* on page 41
- *Network Manager toolbar commands* on page 42
- *Opening and closing the Network Manager pane* on page 42
- *Configuring and connecting to KMDigital controllers and networks* on page 573
- *Configuring the Protocol Gateway service* on page 651

Removing a service

To remove the site connection to a KMC Connect Building Service, select the service in the Network Manager list and then do one of the following:

- On the Network Manager toolbar, click the Delete icon .
- Right-click the service and choose **Delete Service** from the shortcut menu.

Note: Removing a service only removes the service connection to the site database. It does not uninstall it from the computer hosting the service.

Related topics

- *Using the Network Manager* on page 41
- *Network Manager toolbar commands* on page 42
- *Opening and closing the Network Manager pane* on page 42

- *Adding a service* on page 44
- *Configuring an existing service* on page 46

Changing the Network Manager options

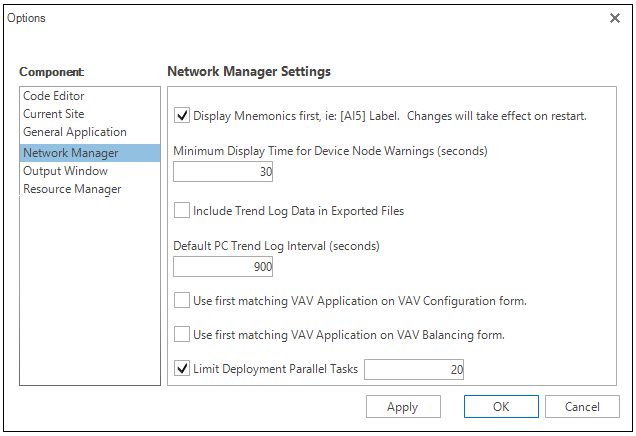
Use the Network Manger Settings dialog to change any of the following settings.

- How objects, points, and tooltips are displayed in the Network Manager list
- Options for trend logs
- Configuration, deployment, and balancing options

To make changes to Network Manager settings, do the following:

1. On the ribbon, click the **File** tab.
2. Click **Options**.
3. From the **Component** list, choose **Network Manager** and then change any of the properties in the dialog.

Illustration 4–7 Network Manager Settings dialog



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, the mnemonic identifier for a point or object is displayed first in the Network Manager list.
- When cleared, the name or label for a point or object is displayed first in the Network Manager list.

Minimum Display Time for Device Node Warnings (seconds) Sets the minimum period—in seconds—that warning icons and tooltips remain in place when hovering 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 117 and *Configuring KMD PC trend logs* on page 125.

Use First Matching VAV Application on VAV Configuration Form When the check box is selected, the application in the newest version of the application library is used. If the

check box is cleared, a list of versions is opened for selection. See the topic *Application selection, configuration, and deployment* on page 655.

Use First Matching VAV Application on VAV Balancing Form When the check box is selected, the application in the newest version of the application library is used. If the check box is cleared, a list of versions is opened for selection. See the topic *Application selection, configuration, and deployment* on page 655.

Limit Deployment Parallel Tasks When the check box is selected, the number of controllers that KMC Connect is actively deploying to is limited by the entered value. This prevents overloading the network. See the topic *Application selection, configuration, and deployment* on page 655.

Related topics

- *Using the Network Manager* on page 41
- *Network Manager toolbar commands* on page 42
- *Opening and closing the Network Manager pane* on page 42

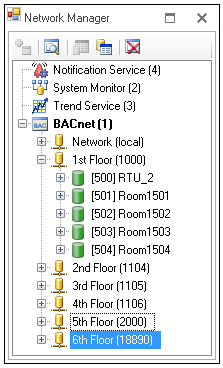
Naming BACnet networks

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 **OK** when finished.

Illustration 4–8 Network Manager with network names



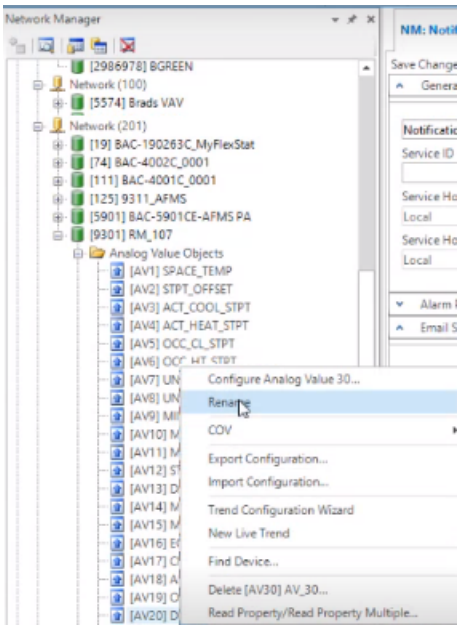
Related topics

- *Using the Network Manager* on page 41
- *Network Manager toolbar commands* on page 42
- *Opening and closing the Network Manager pane* on page 42

Renaming BACnet objects in the Network Manager

BACnet objects may be renamed from within the Network Manager list.

Illustration 4–9 Renaming an object in the Network Manager list



To rename a BACnet object from within the Network Manager list:

1. Right-click the object.
2. Choose **Rename** from the shortcut menu.
3. Enter the new name.
4. Click **OK** when finished.

Related topics

- *Using the Network Manager* on page 41
- *Network Manager toolbar commands* on page 42
- *Opening and closing the Network Manager pane* on page 42

Opening KMC Conquest device served web pages

The Network Manager can be used to open the served web pages of Ethernet-enabled KMC Conquest controllers (those with an "E" suffix in the name, e.g., BAC-5901C**E**). and the BAC-5051(A)E router*.

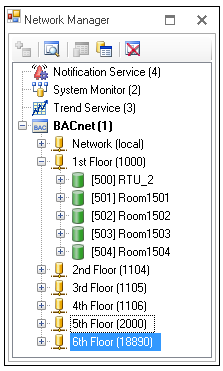
To open the served web pages for a KMC Conquest "E" designated device on the network, do the following:

1. Right-click the network icon .
2. Right-click the device.
3. From the shortcut menu, click **View in Browser**. The page opens in a new browser window.

4 Click **OK** when finished.

For information on working with KMC Conquest controller served web pages, refer to the Conquest Ethernet Controller Configuration Web Pages Application Guide. For information on working with BAC-5051(A)E router served web pages, see the BAC-5051AE Application Guide. For applications involving the BAC-5051(A)E and the Airflow Measurement System (AFMS), refer to the AFMS BAC-5051(A)E Router Application Guide.

Illustration 4–10 Network Manager with network names



Related topics

- *Using the Network Manager* on page 41
- *Network Manager toolbar commands* on page 42
- *Opening and closing the Network Manager pane* on page 42

Restarting BACstac from the Network Manager

For various reasons, 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.

3 From the shortcut menu, choose **Restart BACstac**.

Related topics

- *Using the Network Manager* on page 41
- *Network Manager toolbar commands* on page 42
- *Opening and closing the Network Manager pane* on page 42

Enable and disable NFC

The Near Field Communications (NFC) for KMC Conquest controllers on a network or service can be disabled from either a BACnet network or service in the Network Manager list. This applies only to KMC controllers with the NFC icon and firmware E1.0.0.31 or later.

To enable or disable NFC in a single controller, see the topic *Device objects* on page 443.

Illustration 4–11 The NFC icon



NFC is used with the smartphone app *KMC Connect Lite* to configure Conquest series controllers.

To disable NFC for controllers on a service or network, do the following:

1. Do one of the following:
   - In the Network Manager list, locate the service icon .
   - Expand the Network Manager list, and locate a network icon .
2. Right-click the icon.
3. Choose the NFC shortcut menu.
4. From the NFC shortcut menu, choose **Disable All**.

The NFC on the selected service or network will be disabled.

To enable NFC for controllers on a service or network, do the following:

1. Do one of the following:
   - In the Network Manager list, locate the service icon .
   - Expand the Network Manager list, and locate a network icon .
2. Right-click the icon.
3. Choose the NFC shortcut menu.
4. From the NFC shortcut menu, choose **Enable All** .

The NFC on all controllers on the selected service or network will be enabled.

Section 5: The Output Window

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

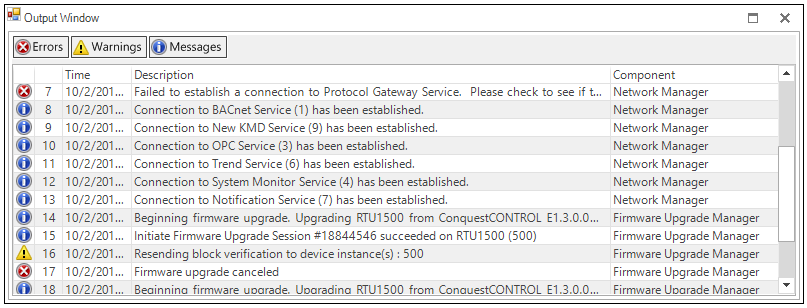
As components of KMC Connect perform tasks, the success or failure of the function is indicated by an entry in the Output Window. Every entry is tagged with an icon as an error, warning, or message.

Table 5–1 Output Window icons

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

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

Illustration 5–1 Output Window



Entries listed in the Output Window are saved as the file `KMCConnectOutput.log` in the Resource Manger Logs folder.

To conserve memory, the Output Window list is limited to 5000 entries. If there are more than 5000 entries, the oldest entries are deleted. KMC Connect adds a message to the list that older entries are in the `KMCConnectOutput.log` file.

Related topics

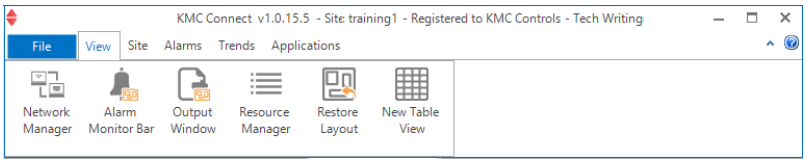
- *Opening and closing the Output Window* on page 54
- *Sorting and clearing the Output Window* on page 54
- *Output Window options* on page 54

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**.

See also the related topics *Sorting and clearing the Output Window* on page 54 and *Output Window options* on page 54.

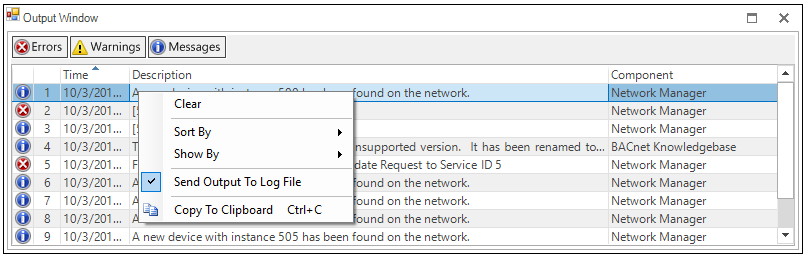
Illustration 5–2 View tab on the ribbon



Sorting and clearing the Output Window

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

Illustration 5–3 Output Window with shortcut menu



Clear Removes all entries from the list.

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

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

Save Output To Log File When selected, entries added to the list are also added to a `.log` text file in the Logs folder in the Resource Manager.

Copy To Clipboard Ctrl+C Selects and copies all of the entries in the list to the Clipboard.

See also the topics Related topics *Opening and closing the Output Window* on page 54 and *Output Window options* on page 54.

Output Window options

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

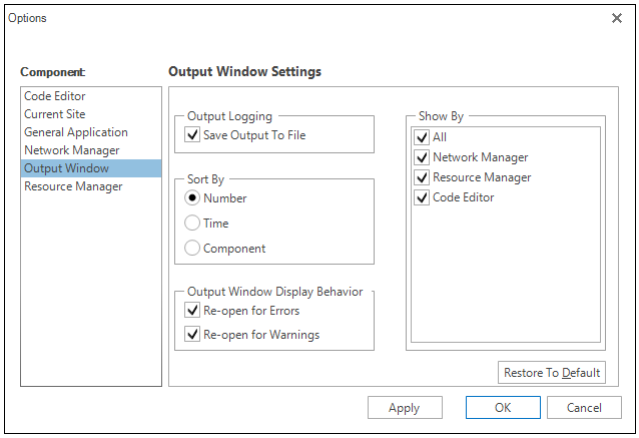
To change the Output Window settings, do the following:

1. On the ribbon click the **File** tab and click **Options**.
2. Click **Output Window**.

3 Change any of the settings.

4 Click **Apply** or **OK** when finished.

Illustration 5–4 Output Window Settings dialog



Output Logging When Save Output to File is selected, entries added to the Output Window are also added to a `.log` text file in the Logs folder of the Resource Manager.

Sort By This property sets the default sort order for the entries in the Output Window.

Output Window Display Behavior When selected, the Output Window automatically opens when error or warning entries are added.

Show By Selects the entries that are visible based on the origin of the message.

Restore to Default Returns the Output Window display to the default settings.

See also the topics *Opening and closing the Output Window* on page 54 and *Sorting and clearing the Output Window* on page 54.

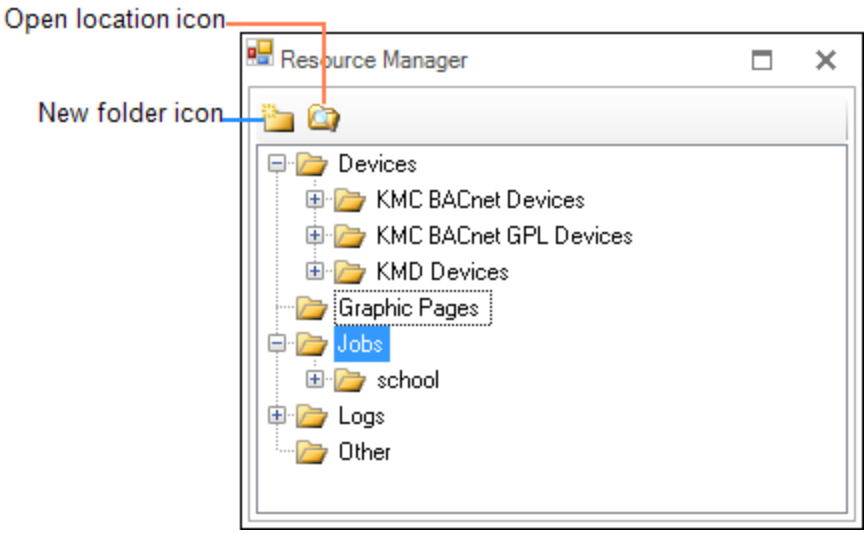
Section 6: About the Resource Manager

The 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 the Resource Manager to create backup files, restore the configuration of controllers with backup files, and edit files offline.

The Resource Manager displays and manages a list of files stored on the computer that is running KMC Connect. Through the 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
- Drag objects into 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

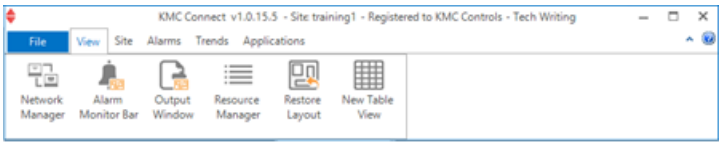
- *Opening and closing the Resource Manager pane* on page 58
- *Editing items in the Resource Manager list* on page 58
- *Restoring a configuration* on page 59
- *Creating backup files* on page 62

- *Managing the Resource Manager list* on page 62
- *Adding files to the Resource Manager folder* on page 63
- *Backing up and restoring BACnet and KMDigital devices* on page 425

Opening and closing the Resource Manager pane

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

Illustration 6–2 View tab on the ribbon



Related topics

- *About the Resource Manager* on page 57
- *Editing items in the Resource Manager list* on page 58
- *Restoring a configuration* on page 59
- *Creating backup files* on page 62
- *Managing the Resource Manager list* on page 62
- *Adding files to the Resource Manager folder* on page 63
- *Importing and exporting BACstage files* on page 428

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

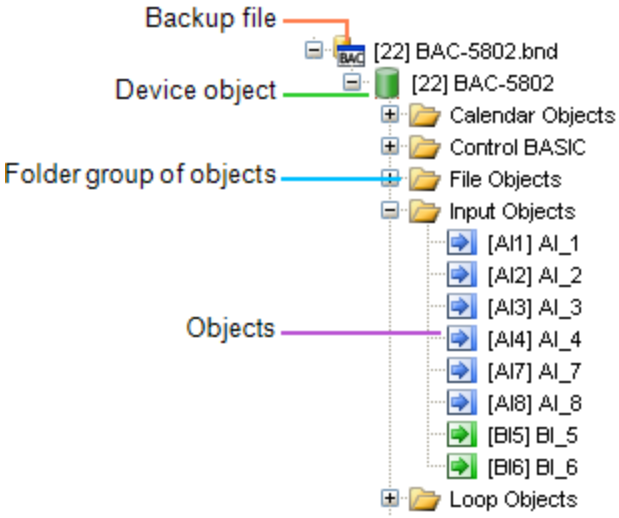
- *About the Resource Manager* on page 57

- *Opening and closing the Resource Manager pane* on page 58
- *Restoring a configuration* on page 59
- *Creating backup files* on page 62
- *Managing the Resource Manager list* on page 62
- *Adding files to the Resource Manager folder* on page 63
- *Importing and exporting BACstage files* on page 428

Restoring a configuration

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

Illustration 6–3 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.

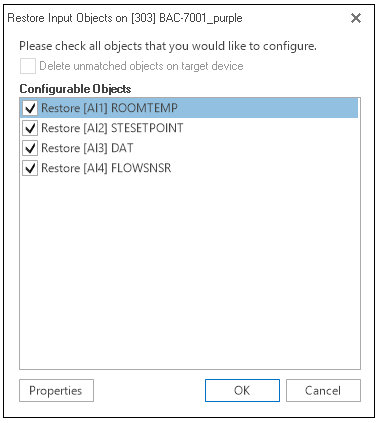
Dragging a folder to another folder or device

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.

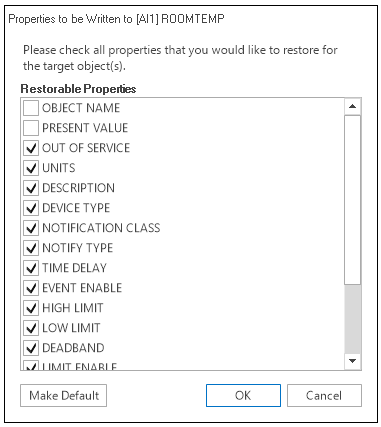
Illustration 6–4 Select Objects dialog



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 6–5 Select properties dialog

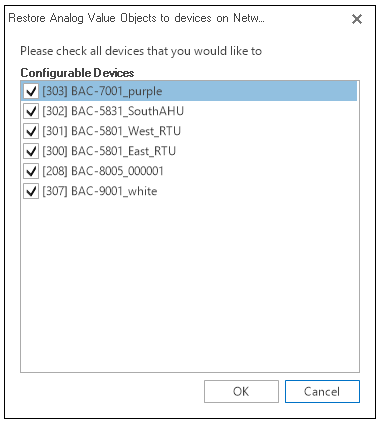


Tip: You may also right-click anywhere in the dialog and then select **Check All** or **Clear All** from the shortcut menu.

Dragging folders to networks

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

Illustration 6–6 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 the Resource Manager* on page 57
- *Opening and closing the Resource Manager pane* on page 58
- *Editing items in the Resource Manager list* on page 58
- *Managing the Resource Manager list* on page 62
- *Adding files to the Resource Manager folder* on page 63
- *Importing and exporting BACstage files* on page 428

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.
- For KMD controllers and points, a `.kmd` file is placed in the Resource Manager list.

Related topics

- *About the Resource Manager* on page 57
- *Opening and closing the Resource Manager pane* on page 58
- *Editing items in the Resource Manager list* on page 58
- *Restoring a configuration* on page 59
- *Managing the Resource Manager list* on page 62
- *Adding files to the Resource Manager folder* on page 63
- *Backing up and restoring BACnet and KMDigital devices* on page 425

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 63.

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.

Renaming an object in the Resource Manager list

1. Right-click the object.
2. Choose **Rename** from the shortcut menu.
3. Enter the new name.
4. When finished, click **OK**.

Duplicating an item in the Resource Manager list

1. Right-click 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 the Resource Manager* on page 57
- *Opening and closing the Resource Manager pane* on page 58
- *Editing items in the Resource Manager list* on page 58
- *Restoring a configuration* on page 59
- *Creating backup files* on page 62
- *Adding files to the Resource Manager folder* on page 63
- *Importing and exporting BACstage files* on page 428

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.

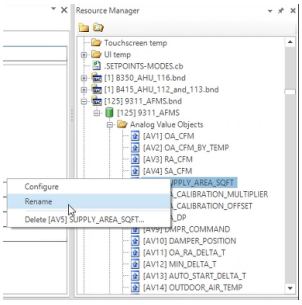
Related topics

- *About the Resource Manager* on page 57
- *Opening and closing the Resource Manager pane* on page 58
- *Editing items in the Resource Manager list* on page 58
- *Restoring a configuration* on page 59
- *Creating backup files* on page 62
- *Managing the Resource Manager list* on page 62
- *Importing and exporting BACstage files* on page 428

Renaming BACnet objects in the Resource Manager

BACnet objects may be renamed from within the Resource Manager list.

Illustration 6–7 Renaming an object in the Resource Manager list



To rename a BACnet object within the Resource Manager list, do the following:

1. Right-click the object.
2. Choose **Rename** from the shortcut menu.
3. Enter the new name.
4. Click **OK** when finished.

Related topics

- *Editing items in the Resource Manager list*
- *Managing the Resource Manager list* on page 62
- *About the Resource Manager* on page 57
- *Opening and closing the Resource Manager pane*

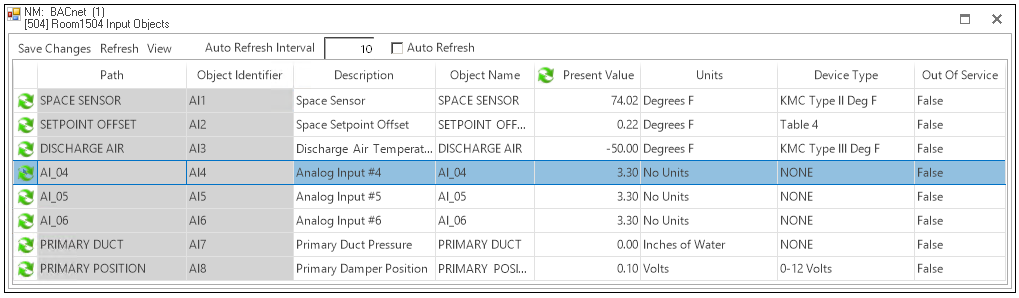
Section 7: Viewing objects with Table View

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.

Use the Table View feature of Design Studio to view all or some of the properties of a group of objects. Table View can be used in two different ways:

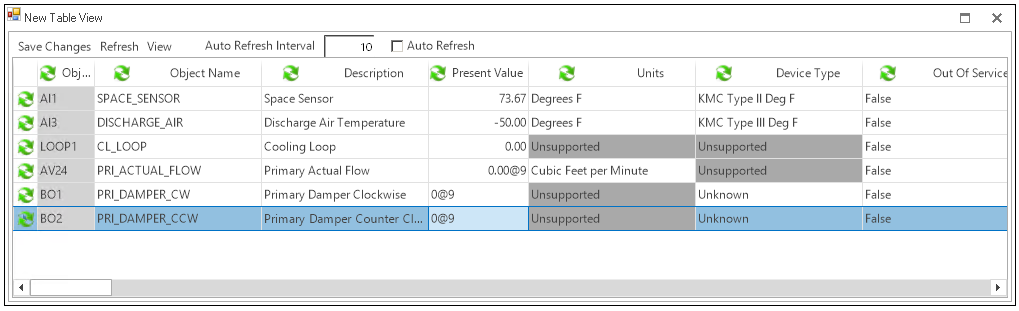
- To view all of the objects in a group folder. See the topic *Viewing groups of similar objects* on page 66.
- As a custom view with a mix of objects from one or more controllers. See the topic *Custom Table Views* on page 69.

Illustration 7–1 Table View of input points



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

Illustration 7–2 Custom Table View



Obj...	Object Name	Description	Present Value	Units	Device Type	Out Of Service
AI1	SPACE_SENSOR	Space Sensor	73.67	Degrees F	KMC Type II Deg F	False
AI3	DISCHARGE_AIR	Discharge Air Temperature	-50.00	Degrees F	KMC Type III Deg F	False
LOOP1	CL_LOOP	Cooling Loop	0.00	Unsupported	Unsupported	False
AV24	PRI_ACTUAL_FLOW	Primary Actual Flow	0.00@9	Cubic Feet per Minute	Unsupported	False
BO1	PRI_DAMPER_CW	Primary Damper Clockwise	0@9	Unsupported	Unknown	False
BO2	PRI_DAMPER_CCW	Primary Damper Counter Cl...	0@9	Unsupported	Unknown	False

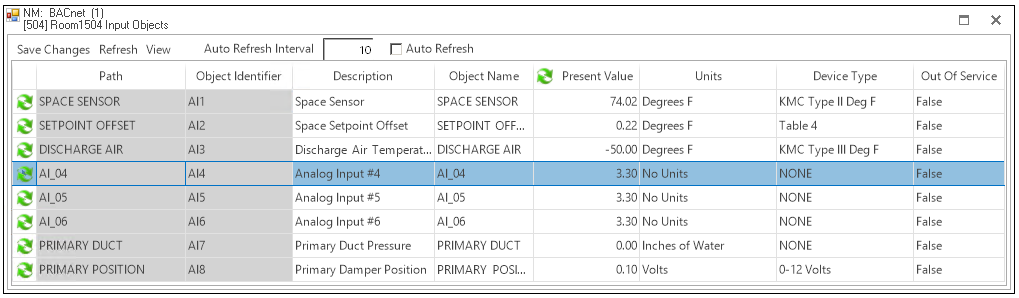
Viewing groups of similar objects

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



Objects, located in the rows of the tables, and properties, located in the columns of the table, can be 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.

To copy table data, do the following.

1. Drag across the rows and columns.
2. Right-click the selected area.

3 From the drop-down menu, choose **Copy Selection to Clipboard**.

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

5 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 Objects**.

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 Properties**.

2 When the Select Visible Properties 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 Objects**. 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 Properties**. 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 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 included in the Table View.

To add objects from the same device or other devices, 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 69.

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.

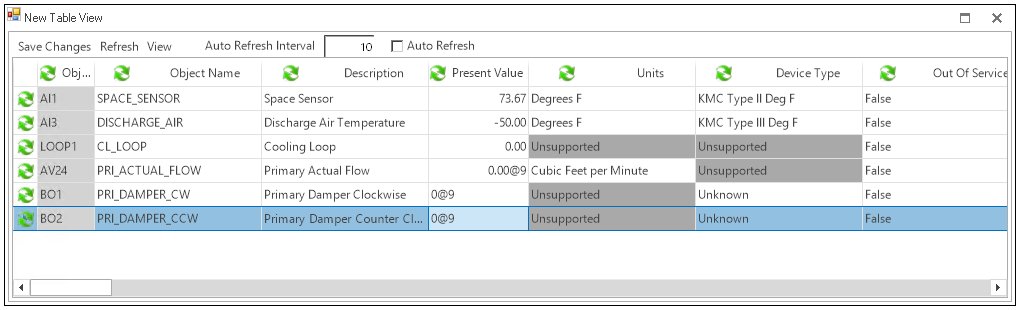
Related topics

- *Viewing objects with Table View* on page 65
- *Custom Table Views* on page 69
- *Using the Network Manager* on page 41

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

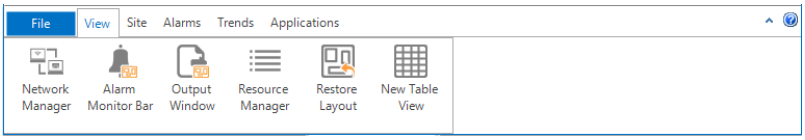


Objects, located in the rows of the tables, and properties, located in the columns of the table, can be moved or hidden to customize the view.

- 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 On the **View** tab, click **New Table View**.



Option: 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.

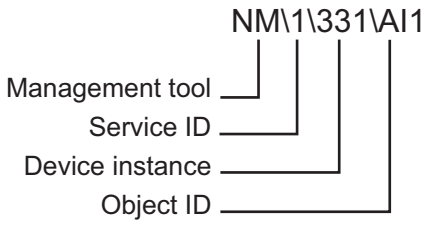
To start a custom Table View from a group of objects, do the following:

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 or points 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.

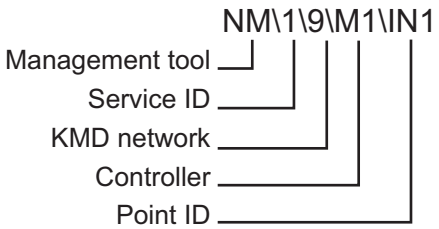
Adding objects by path name

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, several 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 following illustration for path names.



Path for BACnet protocol



Path for KMD protocol

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.

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.

To copy table data, do the following.

1. Drag across the rows and columns.
2. Right-click the selected area.
3. From the drop-down menu, choose **Copy Selection to Clipboard**.
4. Open a document in an application such as spreadsheet, text editor, or word processor.
5. 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 Objects**.
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 Properties**.
2. When the Select Visible Properties 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 Objects**. 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 Properties**. 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 65
- *Viewing groups of similar objects* on page 66
- *Using the Network Manager* on page 41

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 73
- *Restoring site data* on page 74

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

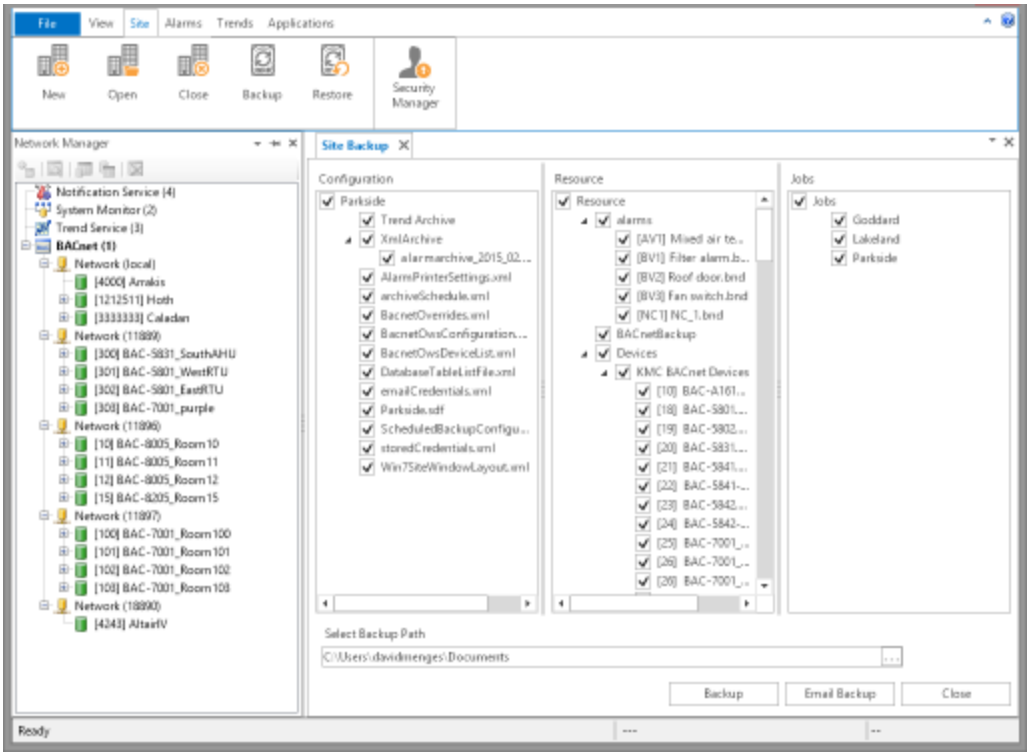
Even though `.zip` files are compressed, the backup file can take up several gigabytes of disk space. The file name is a combination of the site name and the date and time of the backup. Backing up a site may take several minutes to complete.

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.

Site Backup properties

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 74.

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 73.

Section 9: Security and passwords

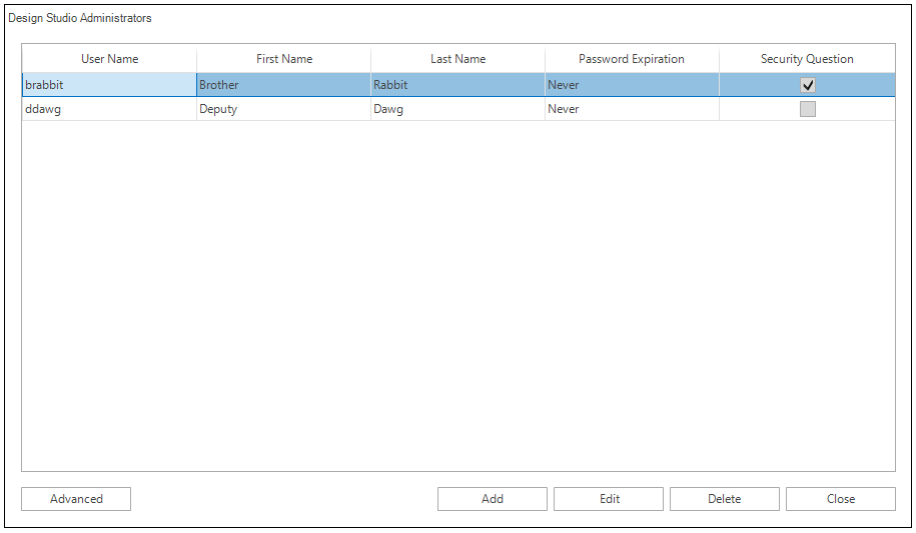
This section covers procedures to manage user access to KMC Connect-managed sites.

Control technicians and engineers use the KMC Connect Security Manager to set up user names, passwords, and TotalControl security groups.

Basic Security With KMC Connect basic security, users are added as site administrators. Most users will use basic security with KMC Connect. For details on managing users with basic security, see the topic *KMC Connect Basic Security* on page 77.

Advanced Security With the advanced security features, 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 *KMC Connect advanced security* on page 80 for details.

Illustration 9–1 KMC Connect basic security dialog



KMC Connect Basic Security

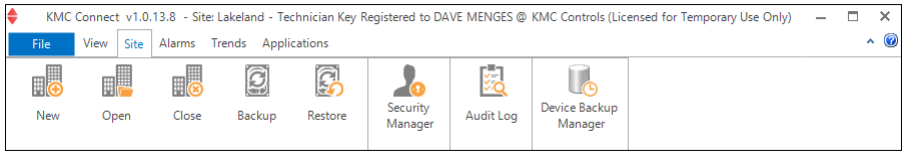
With KMC Connect Basic Security, users are added as site administrators. Most users will use Basic Security with KMC Connect.

Note: Basic Security is the same level of security required for access to TotalControl Design Studio. See *KMC Connect advanced security* on page 80.

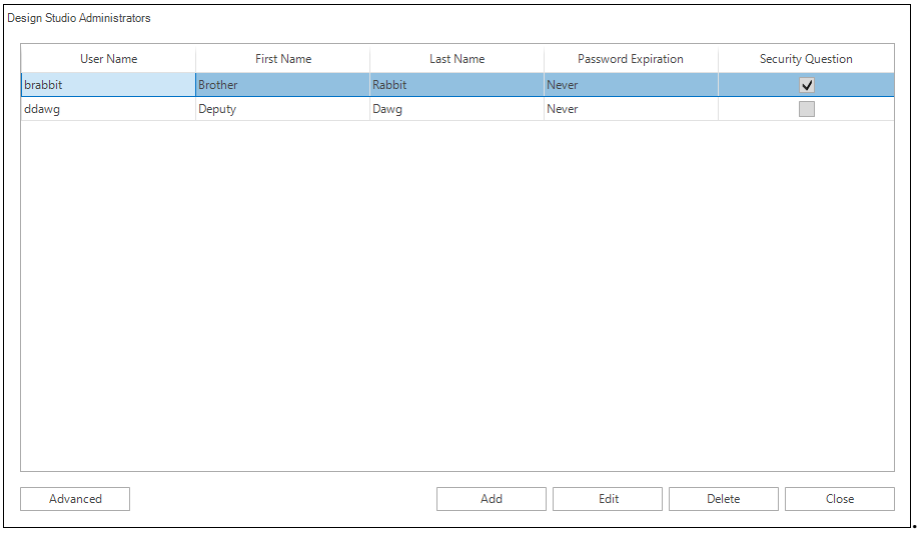
Adding a user

To add a new user, do the following:

1 On the ribbon, choose **Security Manager** from the **Site**. The Design Studio Administrators dialog opens.



2 Click **Add**.



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

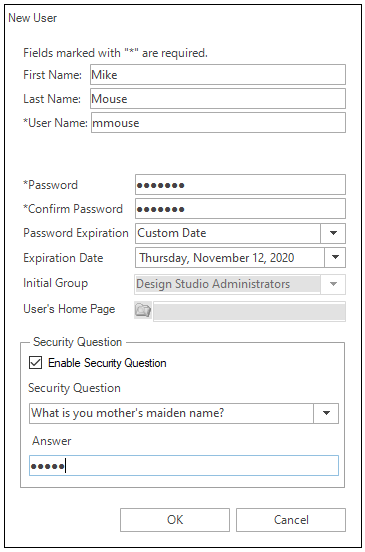
- First and Last Name
- User Name
- Password and confirmation password
- Password expiration date

4 Do the following to add an optional security question:

a. Select the **Enable Security Question** check box.

b. Choose a question from the **Security Question** list.

c. Enter the required response in **Answer**.



5 Click **OK** when finished.

Changing a password

To change a user password, do the following:

1 On the ribbon, choose **Security Manager** from the **Site**. The Design Studio Administrators dialog opens.

2 Select a user from the users list.

3 Click **Edit**.

4 Enter a new password in **Password** and **Confirm Password**.

5 Click **OK** when finished.

Deleting a user

To delete a user, do the following:

1 On the ribbon, choose **Security Manager** from the **Site**. The Design Studio Administrators dialog opens.

2 Select a user from the users list.

3 Click **Remove**.

4 Click **OK** when finished.

KMC Connect Basic Security features

The properties in the Edit User dialog control a user's access to a site.

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 entry in 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.

Initial Group Not applicable to KMC Connect.

User's Home Page Not applicable to 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 enable a security question, select the **Enable Security Question** check box.

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

Answer Enter the word or phrase a user must correctly enter to the security question.

Advanced Click to open the Advanced security dialog.

KMC Connect advanced security

In KMC Connect Advanced Security, users are added to TotalControl security groups. Use the features in Advanced Security if the KMC Connect site will be used to build a TotalControl site.

- *Group roles for security* on page 81
- *Adding TotalControl users to a KMC Connect site* on page 86
- *Adding and modifying groups* on page 83

Group roles for security

KMC Connect security is based on permissions granted to security groups. Users are then assigned to a group based on the level of permission that they need to operate the site. The group is assigned one of the following security roles:

- Admin
- Administrator
- Web Administrator
- Contributor
- Reader

Admin A group with the Admin role is automatically established during installation and the user Admin is automatically added to the group. No other users can be added to this group. The Admin user has complete access to the KMC Connect site.



Tip: KMC Controls recommends that only the highest level site managers have access to the Admin user name and password. The Admin user should establish an Administrator group and site management users should be added to that group.

Administrator User groups in the Administrator role are the only users with access to Design Studio. Operators are granted Administrator status by the Admin user in the Admin group or by other users in the Administrator group. In addition to having access to all functions in Design Studio, a user in the Administrator group can do the following in the Web Portal:

- Add new users to groups.
- Add or move existing users to groups.
- Change the password of existing users.
- View pages except pages that have been denied to the group.
- View graphic elements on a page except those that have been denied to the group.
- Use interactive controls and navigation buttons except those controls that are denied to the group.
- Use all web modules.



Note: Users added with KMC Connect Basic Security are automatically added to the Administrator group.

Web Administrator User groups in the Web Administrator role typically given access to all or major portions of the site, but only through the Web Portal. A user assigned to a Web Administrator group may do the following in the Web Portal:

- Add new users to groups to which the Web Administrator belongs except groups with the Administrator role.
- Add existing users to groups to which the Web Administrator belongs.
- Change the password of existing users except users assigned to an Administrator group and the user logged in as a Web Administrator.
- View pages except pages that have been denied to the group.
- View graphic elements on a page except those that have been denied to the group.
- Use interactive controls and navigation buttons on a page except those controls that are denied to the group.
- Use the web modules assigned to the group including the Web Administer module.

Contributor Users in a Contributor role have limited access to the site through the Web Portal. A user assigned to a contributor group may do the following:

- View pages except pages that have been denied to the group.
- View graphic elements on a page except those that have been denied to the group.
- Use interactive controls and navigation buttons except those controls that are denied to the group.
- Use only the web modules assigned to the group.

Reader The Reader role is the most restrictive security role. A user assigned to a Reader group may do the following in the Web Portal:

- View pages except those that have been denied to the group.
- View graphic objects on a page except those that have been denied to the group.
- Use navigation buttons except those that have been denied to the group.
- Use only the web modules assigned to the group.

Illustration 9–2 Permission roles for groups

	Design Studio	Web access	View web pages	View/change controls	Web modules						
					Web Admin	Trend View	Trend Mgmt	Schedule View	Report View	Alarm View	Timed Overrides
Administrator	◆	◆	◆	◆	◆	◆	◆	◆	◆	◆	◆
Web Administrator		◆	◆	◆	◆	By group permission only [4]					
Contributor		◆[1]	◆[2]	◆[2]							
Reader		◆[1]	◆[3]								

[1]Users must be in a Contributor or Reader group to view graphic pages in a browser.
[2]Users in Contributor groups can view all pages and elements except those denied to a group to which they are assigned.
[3]Users in Reader groups can view all pages except those denied to a group to which they are assigned.
[4]Groups can be denied access to specific trend logs, alarms and schedules.

Related topics

- For an overview of security, see *Security and passwords* on page 77.
- For setting up security for KMC Connect only, see the topic *KMC Connect Basic Security* on page 77.
- For setting up security to use with a TotalControl site, see the topic *KMC Connect advanced security* on page 80.

Adding and modifying groups

Create new security user groups and configure existing groups from the Security Manager Groups tab. Use the Groups tab to do the following:

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

Only operators assigned to a group with the role of Admin or TC Admin can connect to a site with Design Studio.

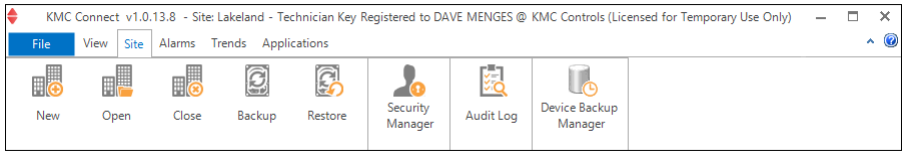
- A security user group consists of one or more users.
- Security roles are assigned to groups.
- Users are assigned to one or more groups.
- Users can perform tasks based upon the groups to which they belong.
- A user's permissions is the combined permissions granted to all of the groups to which the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

Note: KMC Connect does not use Security Groups. Set up Groups only in KMC Connect sites that will used as a basis for a TotalControl site.

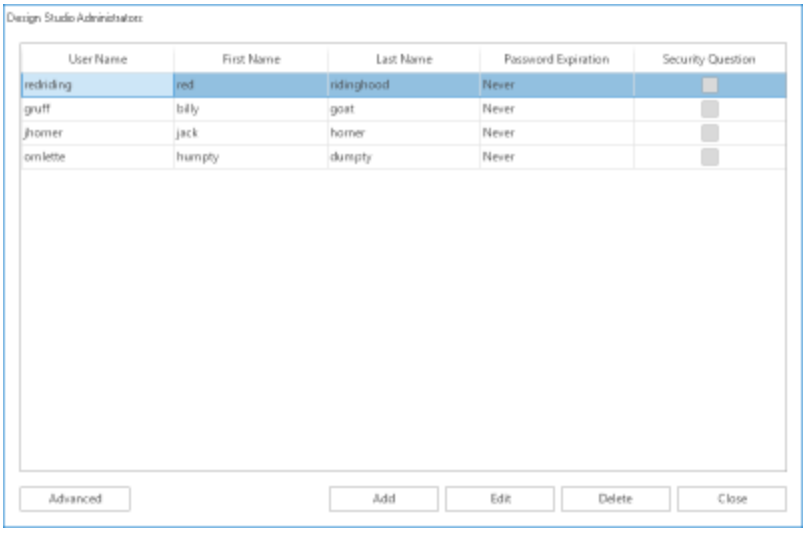
Note: All KMC Connect users must be assigned to a Design Studio Administrator group. All other roles and permissions apply only to TotalControl Web Portal users.

To add a new group, do the following.

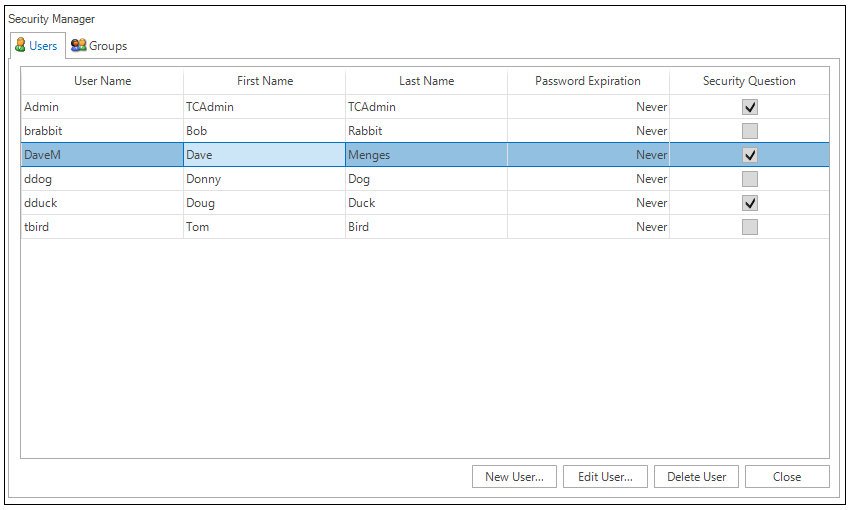
1 On the ribbon, choose **Security Manager** from the **Site** group.



2 Click **Advanced**. The Security Manager dialog opens.



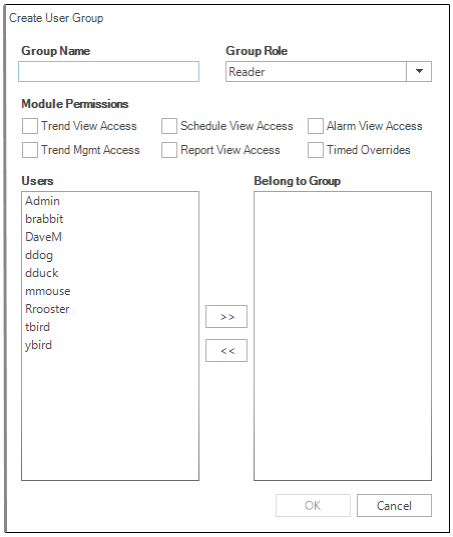
3 In the Security Manager dialog, click the **Groups** tab.



4 Click **New Group**.

5 In the **Create User Group** dialog, enter the following information.

- Enter a name for the new group in **Group Name**.
- Choose a security role from the **Group Role** list.
- In **Module Permissions**, select the optional check boxes next to each web module.



6 If available, add user names to the group.

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

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

7 When finished, click **OK**.

To modify an existing group, do the following.

1 On the ribbon, choose **Security Manager** from the **Site** group.

2 Click **Advanced**.

3 Click the **Groups** tab.

4 Choose a group name from the **Group Name** list.

5 Click **Edit Group**. The **Edit User Group** dialog opens.

6 Add or delete a user or modify the **Group Role** or **Module Permissions** settings.

7 When finished, click **OK**.

To delete an existing group, do the following.

1 On the ribbon, choose **Security Manager** from the **Site** group.

2 Click **Advanced**.

3 Click the **Groups** tab.

4 Choose a group name from the **Group Name** list.

5 Click **Delete Group**.

6 Click **Yes** to confirm or **No** to cancel the deletion.

Adding TotalControl users to a KMC Connect site

Use the KMC Connect Advanced Security Manager to add users and assign them to groups that can be transferred to a TotalControl site with a site backup file. The following information is entered in the Users tab.

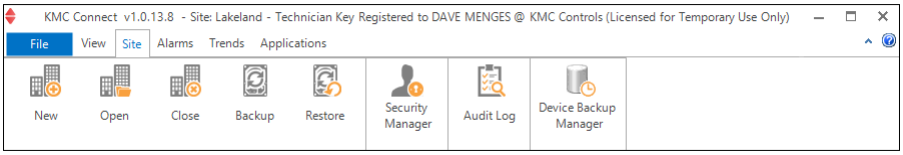
- Names of new users are added.
- User passwords are established and changed.
- If required, a security question is added or changed.
- Users are assigned to security groups.

Users can only be added to existing security groups. See the topic *Adding and modifying groups* on page 83 to add groups to a site.

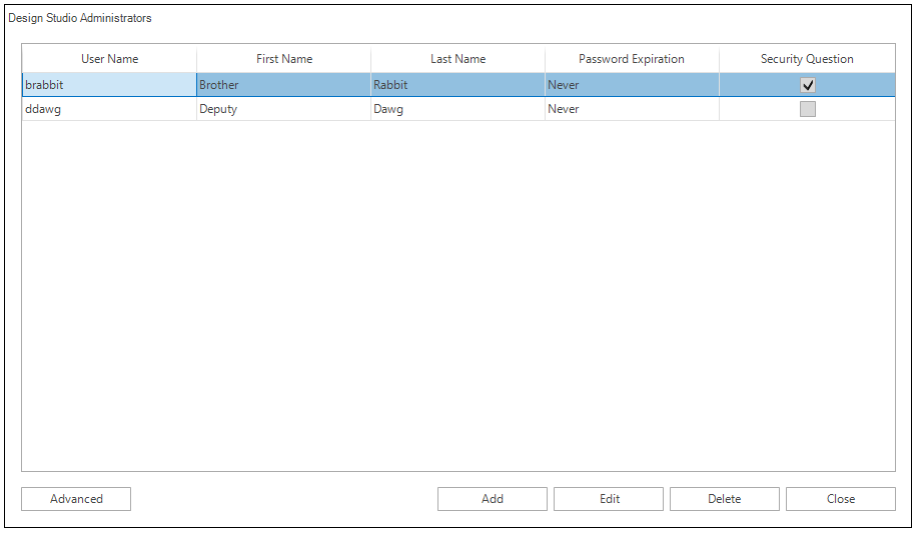
Note: Only users that are assigned to the group Design Studio Administrators have access to a KMC Connect site.

To add a new TotalControl user, do the following:

1 On the ribbon, choose **Security Manager** from the **Site** group.

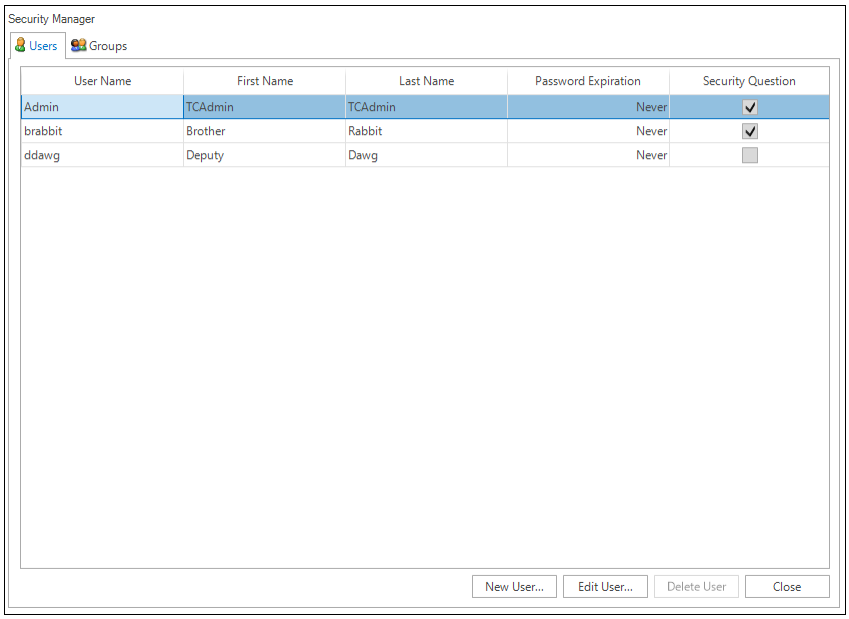


2 Click **Advanced**. The Security Manager dialog opens.



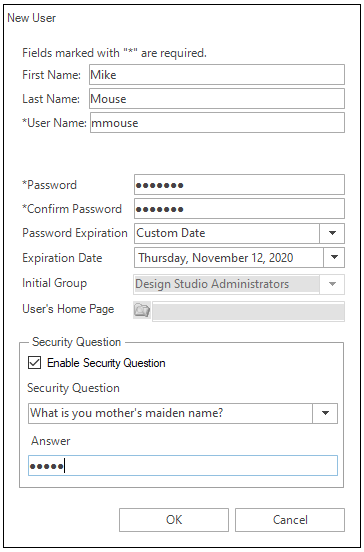
3 In the Security Manager dialog, click the **Users** tab.

4 Click **New User**.



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

- First and Last Name
- User Name
- Password and confirmation password
- Password expiration date



6 Do the following to add an optional security question.

a. Select the **Enable Security Question** check box.

b. Choose a question from the **Security Question** list.

c. Enter the required response in **Answer**.

7 Click **OK** when finished.

Deleting a user

To delete a user, do the following:

1 On the ribbon, choose **Security Manager** from the **Site**.

2 Click **Advanced**.

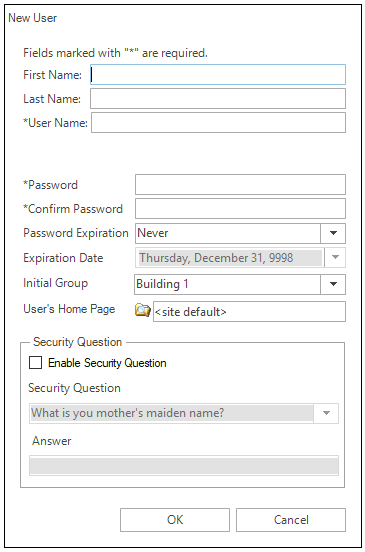
3 Select a user from the users list.

4 Click **Remove**.

5 Click **OK** when finished.

Add Users dialog properties

Illustration 9–3 New User dialog



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 entry in 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 expiration value is "Never".

Expiration Date Sets the date the password will expire.

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.

User's Home Page Sets the page in the TotalControl Web Portal to which the user will be directed after log in.

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.

For an overview of security, see *KMC Connect advanced security* on page 80.

Section 10: The KMC Connect Audit Log

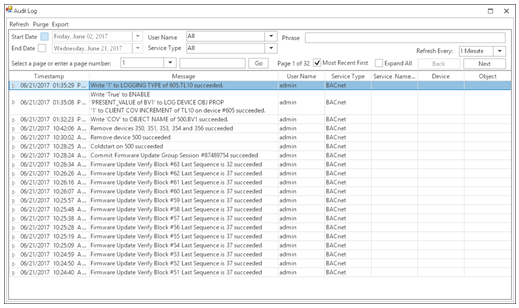
KMC Connect includes an Audit Log that tracks changes to the site.

The Audit Log viewer shows 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 controllers, either BACnet or KMD, 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 **Site** tab, click **Audit Log**. The Audit Log viewer opens.
2. Enter criteria—dates, User Name, Phrase, Service Type—to filter the data listed in the Audit Log.
3. Click **Refresh**.

Exporting the Audit log

The visible data in the Audit Log can be saved in a comma separated values (`.csv`) file.

To export the Audit Log, do the following:

1. Open the Audit Log for viewing.
2. Enter criteria—dates, User Name, Phrase, Service Type—to filter the data listed in the Audit Log.
3. At the top of the log, click **Export**.
4. Browse to a location to save the data.
5. Click **Save**. KMC Connect saves the log with the site name as the file name and `.csv` as the file extension.

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:

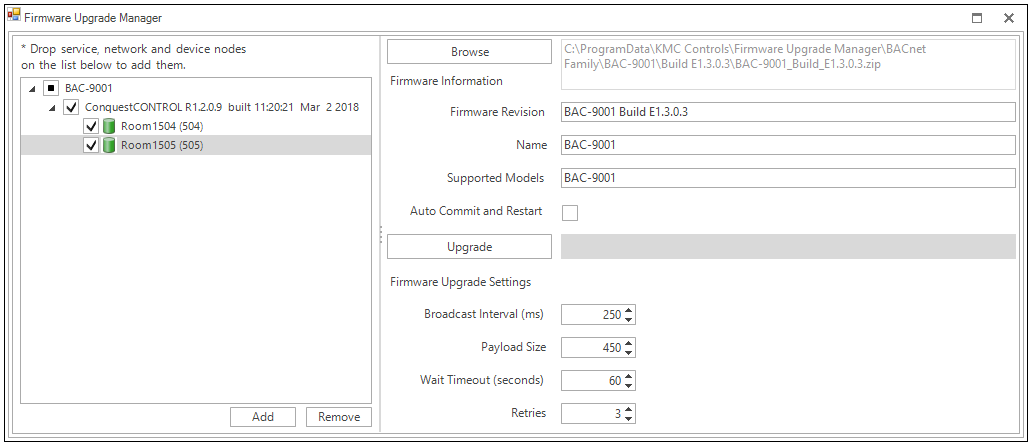
1. Open the Audit Log for viewing.
2. At the top of the log, click **Purge**.
3. In the Purge Audit Log dialog, set the time and date of the oldest record to be retained.
4. Click **OK**.
5. Click **YES** to confirm the purge. Clicking **NO** cancels the operation.

Section 11: Upgrading firmware

This section covers upgrading controller firmware over the network with KMC Connect.

Firmware in later model Conquest series devices can be upgraded—or flashed—over the network. Refer to the release notes and specifications for each device to verify compatibility.

Illustration 11–1 Firmware Upgrade Manager

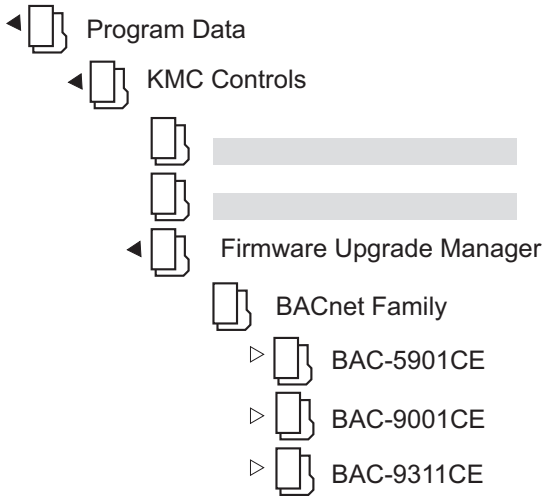


Downloading firmware

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

1. Download the firmware file from the KMC Controls web 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 a self-running file that will install the firmware.

3 Double-click the extracted file. The installer will place 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.



Upgrading firmware

1 Locate the device, network, or service icon in the Network Manager list.

2 Right-click the icon and choose **Upgrade 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 is stored in a folder with the model name for the controller.

5 If the controllers can restart immediately after the upgrade, select **Auto Commit and Restart**.

Caution: Do not select **Auto Commit and Restart** if equipment connected to a controller cannot be shut down and restarted when the upgrade is finished.

6 Click **Upgrade**. KMC Connect will begin loading the new firmware into the selected controllers.

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

- To finish upgrading, select the devices and click **Commit**.
- Click **Abort** to cancel the upgrade 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:

- Reduce the Payload value to reduce the number of rejected packets.
- 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-5051(A)E 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.

Auto Commit and Restart When selected, the Firmware Upgrade Manager will commit the upgrade to the controller and immediately reinitialize the controller with a cold start. Select this feature only if controller operation can be interrupted during reinitialization.

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 the firmware upgrade but will keep the firmware upgrade from interfering with traffic on busy networks.

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.

Wait 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 provide busy devices additional time to handle and respond to firmware upgrade requests.

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 a limited 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 110
- *Configuring BACnet controller trend logs* on page 113
- *Configuring KMD trend logs and groups* on page 121

Viewing trend logs

To view trend logs, see the following topics.

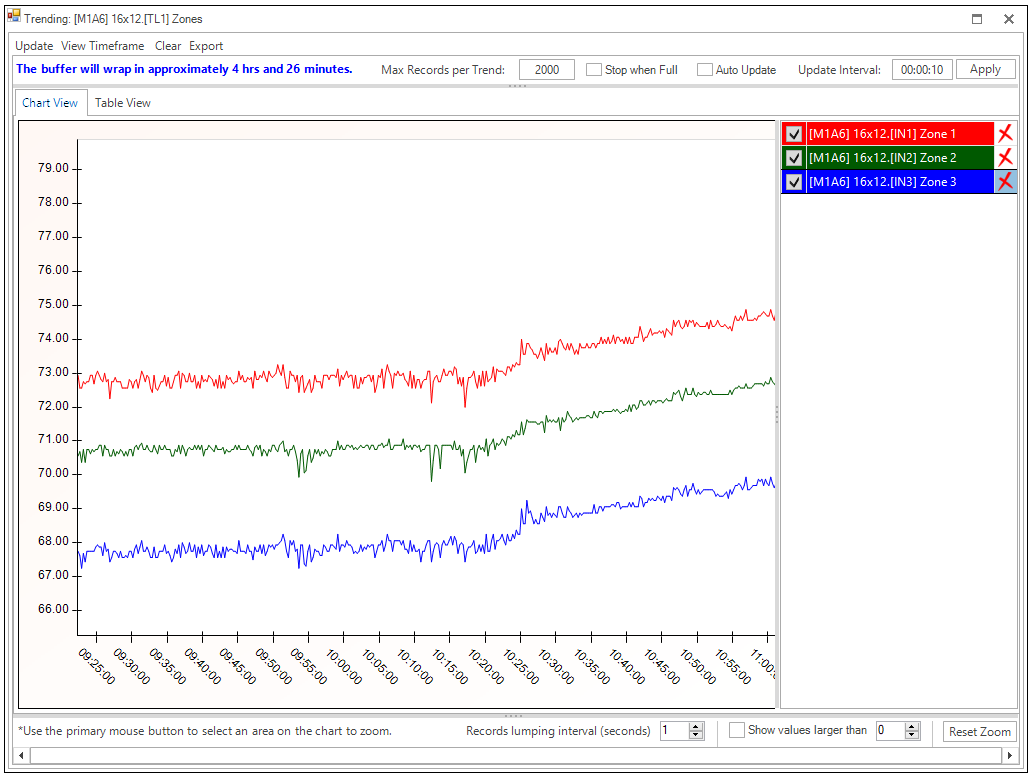
- *Viewing controller trend logs* on page 106
- *Viewing with the Trend Viewer* on page 102

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 106.
- To view temporary data from one or more points or objects, see the topic *Viewing with the Trend Viewer* on page 102 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 trend logs in the Trend Manager*.



Viewing with the Trend Viewer

The Trend 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 102.
- To view device trend logs, PC trend logs, or trend groups, see *Viewing trend logs in the Trend Manager* on page 103.

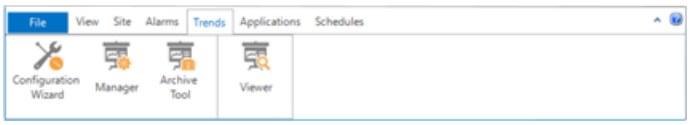
Illustration 13–1 Trend Viewer



Starting a live trend log

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

1 On the ribbon choose **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.

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

- The number of analog objects is limited only by viewing clarity.
- If a mix of analog, binary, loop or multistate objects is 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 from which to collect data in the Network Manager list. This may be an analog, binary, loop, or multistate object.

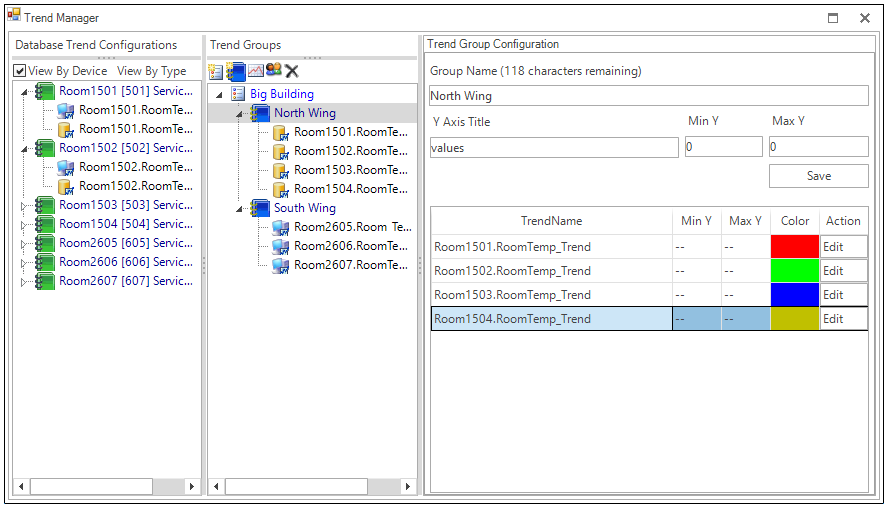
2 Right-click the object and select **New Live Trend** from the shortcut menu.

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

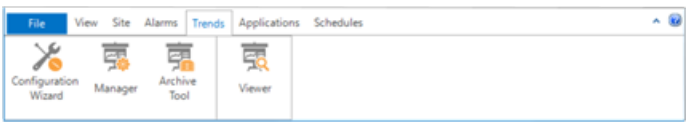
Viewing trend logs in the Trend Manager

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

Illustration 13–2 KMC Connect Trend Manager



1 On the ribbon, choose **Manager** from the **Trends** group. The Trend Manager opens.



2 Do one of the following:

- In the Database Trend Configurations list or Trends Group list, right-click on a PC trend object icon or Device Log icon and choose **Open Trend Viewer** from the shortcut menu.
- In the Trend Groups list, right-click on a Trend Group icon 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.

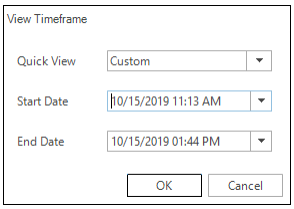
Illustration 13–3 Trend Viewer



Update Click to manually refresh the chart or table with the newest data in the objects.

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

Illustration 13–4 The View Timeframe dialog



Clear Click to remove all trend data from the viewer.

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

Max Records per Trend Sets the maximum number of data points held in the buffer of the Trend Viewer.

Stop when Full When selected and the number of data points of the trend exceeds the value of Max Records per Trend, the Trend Viewer will stop collecting data. If Stop when Full is not selected, the oldest data in the buffer will be deleted.

Auto Update When selected, the data in the Trend Viewer is automatically updated at the interval specified in Update Interval.



Note: In order for Auto Update to function properly, the Device ID for either TotalControl or KMC Connect (depending on which software you are using) must be configured to receive Buffer Ready notifications.

Update Interval Enter a value to automatically refresh the data in the Trend Viewer.

Apply Click to apply the settings for Stop when Full, Max Records per Trend, Auto Update, Update Interval.

Records lumping interval (seconds) When exporting multiple trend logs to a `.csv` file, selecting **Records lumping interval** lines up the records to the nearest specified time value.

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

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 the top left corner of the chart to the bottom right corner. A blue box shows the extent of the area to be viewed.
- To zoom out, drag from either the top right corner of the chart to the bottom left corner or from the bottom right corner to the top left corner. A pink box shows the extent of the area to be viewed.

Reset Zoom Restores the display to the original view.

Related topics

- *Configuring BACnet trend logs and groups* on page 109
- *Viewing controller trend logs* on page 106

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.

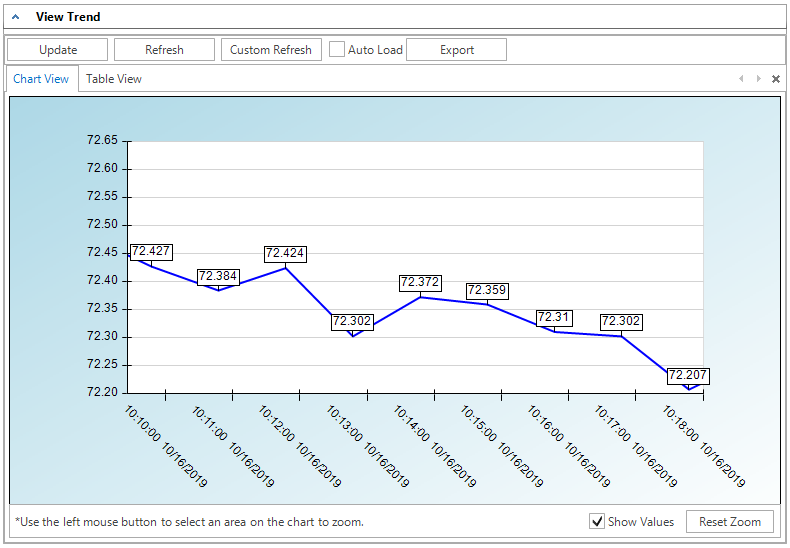
To view controller trend data from a point or object, do the following.

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 scroll to **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 Viewer* on page 102.

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

Illustration 13–5 BACnet controller trend with text displayed



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

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

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

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

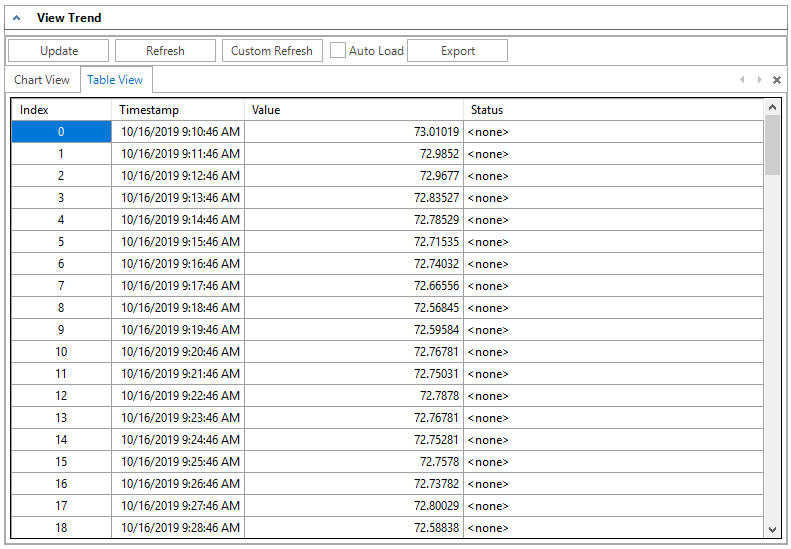
Show Values Select to display the actual recorded text values of the data.

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

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

- To zoom in, drag from the top left corner of the chart to the bottom right corner. A blue box shows the extent of the area to be viewed.
- To zoom out, drag from either the top right corner of the chart to the bottom left corner or from the bottom right corner to the top left corner. A pink box shows the extent of the area to be viewed.
- Click **Reset Zoom** to restore to the original view.

Illustration 13–6 BACnet controller trend —table view



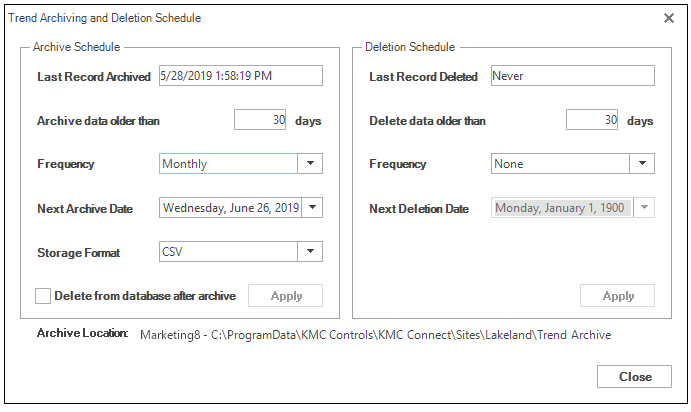
Index	Timestamp	Value	Status
0	10/16/2019 9:10:46 AM	73.01019	<none>
1	10/16/2019 9:11:46 AM	72.9852	<none>
2	10/16/2019 9:12:46 AM	72.9677	<none>
3	10/16/2019 9:13:46 AM	72.83527	<none>
4	10/16/2019 9:14:46 AM	72.78529	<none>
5	10/16/2019 9:15:46 AM	72.71535	<none>
6	10/16/2019 9:16:46 AM	72.74032	<none>
7	10/16/2019 9:17:46 AM	72.66556	<none>
8	10/16/2019 9:18:46 AM	72.56845	<none>
9	10/16/2019 9:19:46 AM	72.59584	<none>
10	10/16/2019 9:20:46 AM	72.76781	<none>
11	10/16/2019 9:21:46 AM	72.75031	<none>
12	10/16/2019 9:22:46 AM	72.7878	<none>
13	10/16/2019 9:23:46 AM	72.76781	<none>
14	10/16/2019 9:24:46 AM	72.75281	<none>
15	10/16/2019 9:25:46 AM	72.7578	<none>
16	10/16/2019 9:26:46 AM	72.73782	<none>
17	10/16/2019 9:27:46 AM	72.80029	<none>
18	10/16/2019 9:28:46 AM	72.58838	<none>

Archiving trend logs

Use the Trend Archive tool to manage the storage of trend data in the KMC Connect site. Deleting data permanently removes the trend data from the site database. Archiving data saves the trend data in either a `.csv` or `.xml` file. Both types of files are saved in the Trend Archive folder located at the following location:

```
C:\ProgramData\KMC Controls\KMC Connect\Sites
\site name\Trend Archive
```

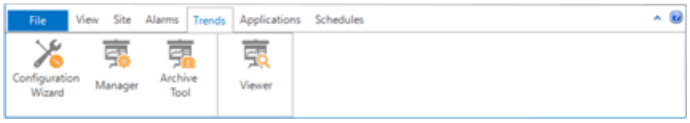
Illustration 13–7 Trend Archiving and Deletion Schedule dialog



Note: KMC Connect must be running when an archive or delete process is scheduled to run.

To archive and optionally delete trend data, do the following:

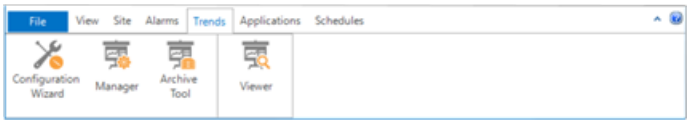
1. On the ribbon, choose **Archive Tool** from the Trends group. The Trend Archiving and Deletion Schedule dialog opens.



2. Under **Archive Schedule**, set the age of the data to archive (and optionally delete) in the **Archive data older than** text box.
3. Select an option from **Frequency**.
   - Choose **IMMEDIATE** to archive or delete data as soon as **Apply** is clicked.
   - Choose from the other options to repeat the action on a regular schedule. Frequency selection ranges from daily to yearly.
4. In **Next Archive Date**, enter a date when the archiving process will next run.
5. Choose either **CSV** or **XML** from **Storage Format**.
6. To permanently remove the data from the site after the data is archived, select **Delete from database after archive**.
7. Click **Apply** when ready.

To delete trend data, do the following:

1. On the ribbon, choose **Archive Tool** from the Trends group. The Trend Archiving and Deletion Schedule dialog opens.



2. Under **Delete Schedule**, set the age of the data to delete in the **Delete data older than** text box.
3. Select an option from **Frequency**.
   - Choose **IMMEDIATE** to delete data as soon as **Apply** is clicked.
   - Choose from the other options to repeat the action on a regular schedule. Frequency selection ranges from daily to yearly.
4. In **Next Deletion Date**, enter a date when the deletion process will next run.
5. Click **Apply** when ready.

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.

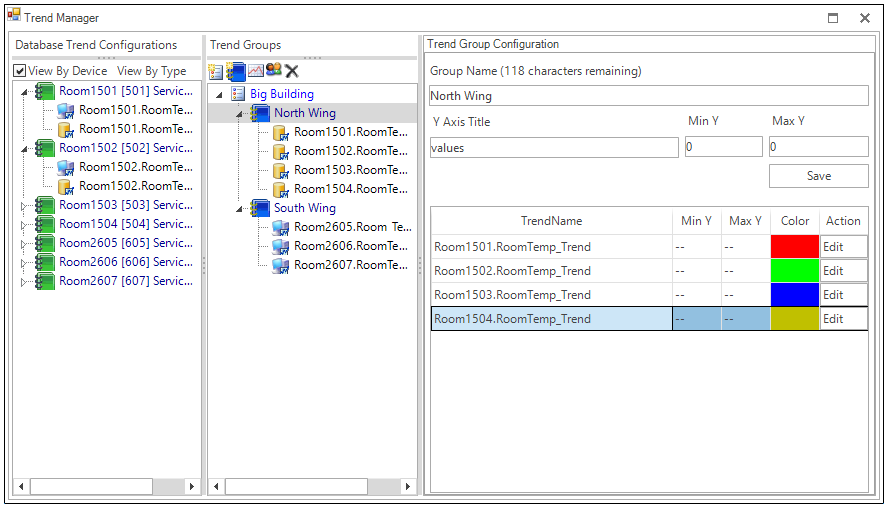
The easiest method for setting up device and controller trend logs is to use the Trend wizard. See *Trend Configuration Wizard* on page 110.

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

- *Configuring BACnet controller trend logs* on page 113
- *Configuring BACnet device trend logs* on page 114
- *Configuring a BACnet PC trend log* on page 117

To view trend data see, *Viewing trend logs in KMC Connect* on page 101.

Illustration 14–1 Trend Manager



Trend Configuration Wizard

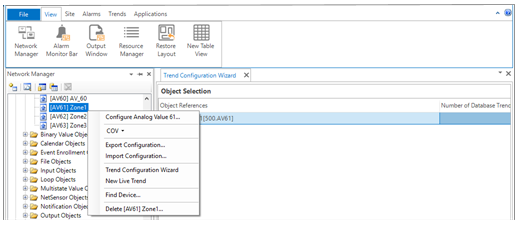
The Trend Configuration Wizard is an easy method to configure trend logs for BACnet controllers. Using the wizard is an alternative to manually setting up the properties for trend log, notification class, and event enrollment objects.

- 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.
- The wizard is an easy way to configure BACnet controller trend logs and use them as a device logs in the Trend Manager.

Start the wizard

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

- On the ribbon, choose **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:

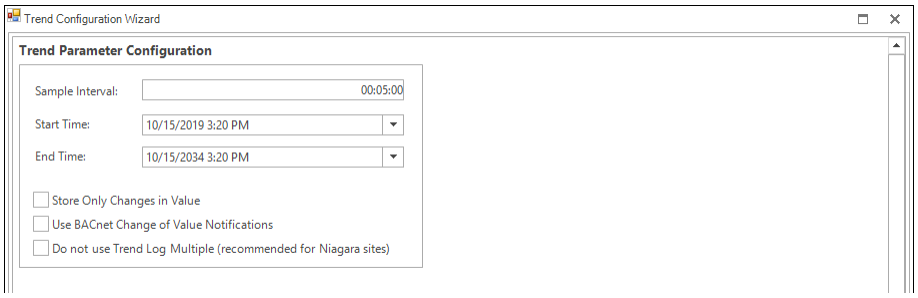
- Drag input, output, or value objects from the Network Manager list to the Object Selection list.
- Right-click an object in the Network Manager list and choose **Trend Configuration Wizard**.
- Click **Add** and then search for objects. See the procedure *Add objects by searching* on page 112.

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

3 When ready, click **Next**.

4 Change or accept the Trend Parameter Configuration properties.

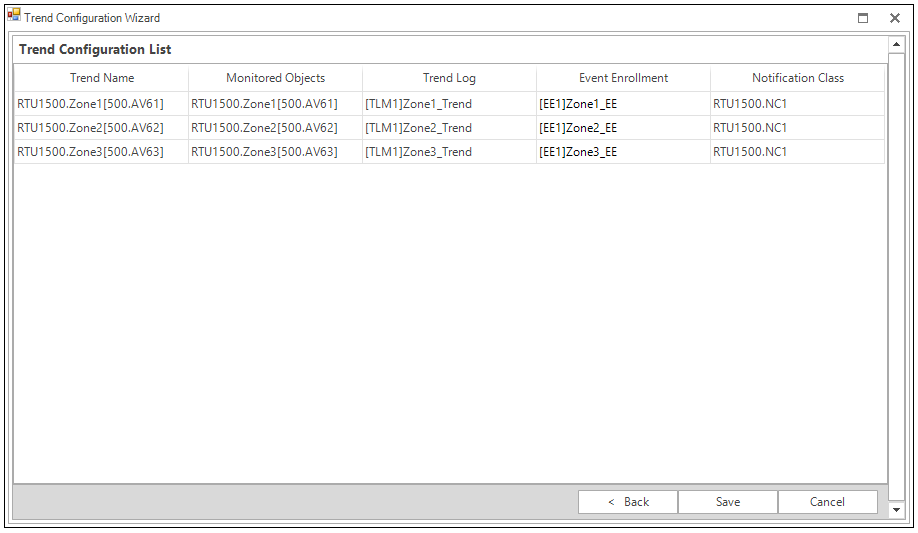
- Sample Interval sets the time between samples.
- **Start** and **End Time** sets the time span for logging data.
- Select the **BACnet Change of Value Notifications** check box to set up a COV subscription. COV subscriptions are not supported in all devices. For additional information, see the topic *Change Of Value (COV) notifications* on page 155.



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.



Tip: If unconfigured Trend Log and Trend Log Multiple objects are both available in the controller, the Wizard chooses a Trend Log Multiple object. To change to a Trend Log object, right-click on the Trend Log object cell and choose another Trend Log object.

7 Click **Save** when finished.

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

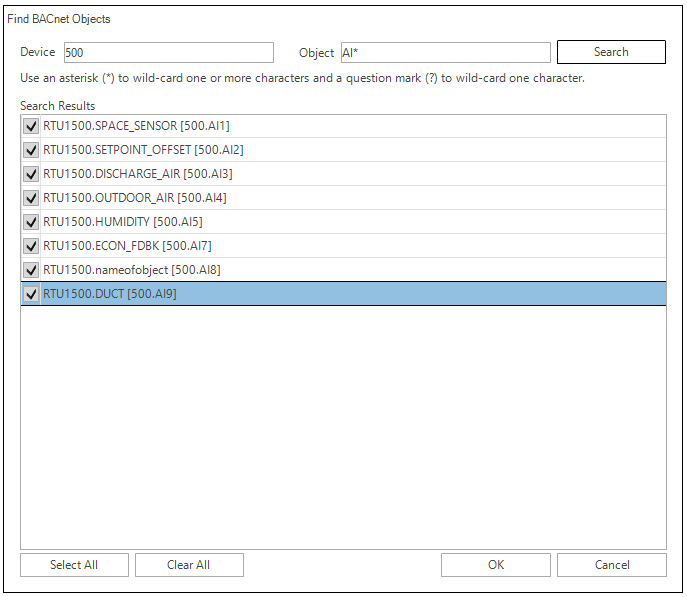
Add objects by searching

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

To add objects by searching, do the following:

1 Start the wizard.

2 In the Object Select list, click **Add**. The Find BACnet Objects Dialog opens.



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 objects* on page 509
- *Notification objects* on page 490
- *Event enrollment objects* on page 451

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

- *Configuring BACnet controller trend logs* on page 113
- *Configuring BACnet device trend logs* on page 114
- *Configuring a BACnet PC trend log* on page 117

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 106.
- 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 119
- To set up trend logs with a wizards, see *Trend Configuration Wizard* on page 110.

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.

   Enter the number of samples that will trigger the notification recipient that there is data in the buffer available for transfer. For device trend logs, a value of 10 provides a good trade-off between responsiveness and excessive network traffic.
8. 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**.

9 Under **Property** in **Log Object Property Reference**, if the property to log is not **Present Value**, choose the correct property.

10 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.

11 Select or clear the **Stop When Full** check box.

12 If a notification class object will be used with this trend log, expand **Event/Alarm Properties** and do the following:

13 In **Notification Class**, enter the object number of the notification class object.

14 Under **Event Enable**, do the following.

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

15 In **Notification Type**, select **Event.**

16 When all selections and entries are correct, click **Save Changes** at the top of the tab.

Related topics

- *About trend logs* on page 99
- *Viewing trend logs in KMC Connect* on page 101
- *Configuring BACnet device trend logs* on page 114
- *Configuring a BACnet PC trend log* on page 117
- *Adding BACnet trend logs to trend groups* on page 119

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 119
- To set up trend logs with a wizards, see *Trend Configuration Wizard* on page 110.

Configuring BACnet trend objects for a device trend log

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

1 Follow the procedure in the topic *Configuring BACnet controller trend logs* on page 113 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 configuring 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 setup 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 116.

5 Enter the trend configuration settings in the dialog.

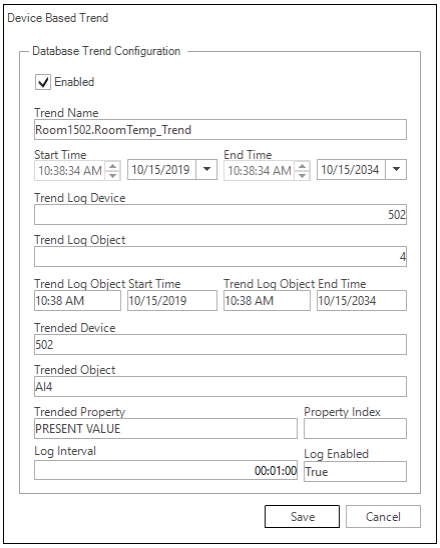
6 Click **Submit** when finished.

7 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.

Illustration 14–2 BACnet Device Based Trend dialog



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.

4 Click **Save** when done.

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 the 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 99.

- An object must be added to a trend group to be visible in the Web Portal.
- 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 119
- To set up trend logs with a wizard, see *Trend Configuration Wizard* on page 110.

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 117.
4. Enter the trend configuration in the dialog.
5. Click **Submit** when finished.
6. Drag the object from the **Device Trend Configurations** list to a trend group in the **Trend Groups** list.

Using the PC Based Trends dialog

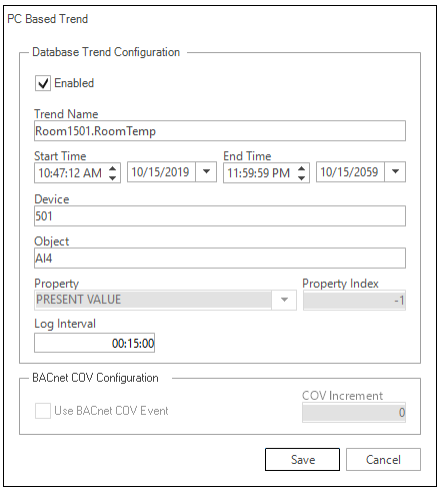
Use the PC Based Trend dialog to set up a PC based trend for a BACnet object. The dialog opens automatically when a BACnet object is dropped into the Database Trends Configuration list.

To open the dialog, do the following:

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



Enabled Select the check box to add the records for an individual point from a BACnet 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 Index (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 48 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 155.

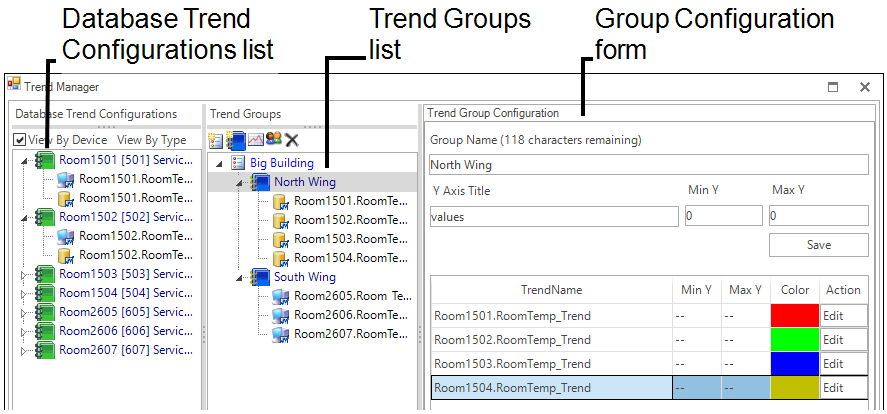
COV Increment (Available only when dragging objects that support change of value to the Device Trend Configuration list.) Enter the value by which **Property** must change to initiate a COV notification.

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 110.
- To view the data with the Trend Log viewer, see *Adding BACnet trend logs to trend groups* on page 119.

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 ConnectTrend 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 list, 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 Manager 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 list 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.

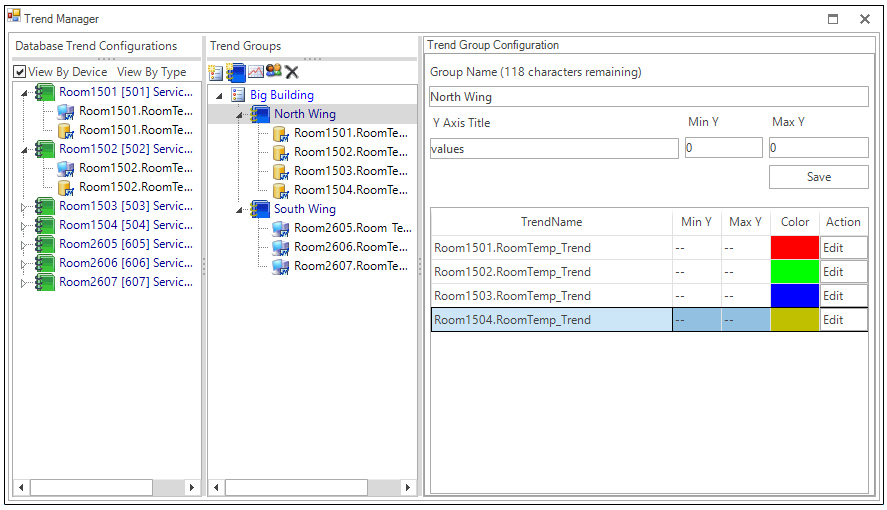
Section 15: Configuring KMD trend logs and groups

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

Set up and manage trend logs with the Trend Manager.

- *About trend logs* on page 99
- *Configuring KMD controller Trend Logs* on page 121
- *Configuring KMD device trend logs* on page 123
- *Configuring KMD PC trend logs* on page 125
- *Viewing trend logs in KMC Connect* on page 101
- *Configuring BACnet trend logs and groups* on page 109

Illustration 15–1 Trend Manager



Configuring KMD controller Trend Logs

A controller Trend Log is the mechanism that KMC Connect uses to collect the data for a device Trend Log.

A controller Trend Log can be set up two different ways.

- To log data at regular intervals, see the procedure *Automatic logging* on page 122.
- To use Control Basic to log a single record at a time, see the procedure *On-demand logging* on page 122

See also the topic *Trend Log points* on page 615 for a detailed description of the properties in the Trend Log point.

Automatic logging

A controller Trend Log can be set up to log one or more points at regular intervals. To set up a Trend Log for automatic logging, do the following:

1. In the Network Manager list, locate the KMD controller and then select the Trend Log icon that will collect the data.
2. Expand **General Properties**.
3. Under **Name**, enter a name for the Trend Log.
4. Under **Log Interval**, enter the frequency to sample data. This is entered as *hours*: *minutes*: *seconds*.
5. Under **Trended Points**, enter the points to be recorded in the format *controller.mnemonic*.
   - Mnemonics are listed in the topic *Mnemonics for KMD controllers* on page 239.
   - The recorded points can be from points within the controller or from other controllers on the network.
   - The exact number of points placed in the list depends on the type of controller.
6. When all selections and entries are correct, click **Save Changes** at the top of the tab.

On-demand logging

A Trend Log can be set up to log data only when triggered by a statement in a Control Basic program. To set up a KMD controller Trend Log for on-demand logging, do the following:

1. In the Network Manager list, locate the KMD controller and then select the Trend Tog icon that will collect the trend data.
2. Expand **General Properties**.
3. Under **Name**, enter a name for the Trend Log.
4. Under **Log Interval**, enter 00:00:00.
5. Under **Trended Points**, enter the points to be recorded in the format *controller.mnemonic*.
   - Mnemonics are listed in the topic *Mnemonics for KMD controllers* on page 239.
   - The recorded points can be from points within the controller or from other controllers on the network.
   - The exact number of points placed in the list depends on the type of controller.
6. When all selections and entries are correct, click **Save Changes** at the top of the tab.

7 Add a Control Basic statement `TLx = -1` to one of the Control Basic programs. See the following example for the details.

Control Basic example The following example adds one record to Trend Log TL1 on the first day of each month.

```
IF+ DOM < 2 THEN TL1 = -1
```

The exact data add to the Trend Log is determined by the Trended Points entry in the Trend Log point.

See also the topic *Programming with the Code Editor* on page 195.

Related topics

- *Trend Log points* on page 615
- *About trend logs* on page 99
- *Configuring KMD device trend logs* on page 123
- *Adding KMD trend logs to a trend group* on page 126

Configuring KMD device trend logs

KMD device trend logs are controller trend logs that have been retrieved periodically from a controller and then stored in the site database. A device trend reduces network traffic and is the preferred method to capture trend log data.

To configure a device trend log for KMD controllers, do the following:

1 Follow the procedure in the topic *Configuring KMD controller Trend Logs* on page 121 to set up a controller trend log.

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

3 Locate the Trend Log icon in the Network Manager list.

4 Drag the trend log from the Network Manager list to the **Database Trend Configurations** list in the Trend Manager. The Device Based Trends dialog opens.

5 In the Device Based Trends dialog, set the following:

- Clear the check box for any trend in the trend log that you do not want to add to the trend group.
- Set dates for the trend to collect data with **Start Time** and **Stop Time.**

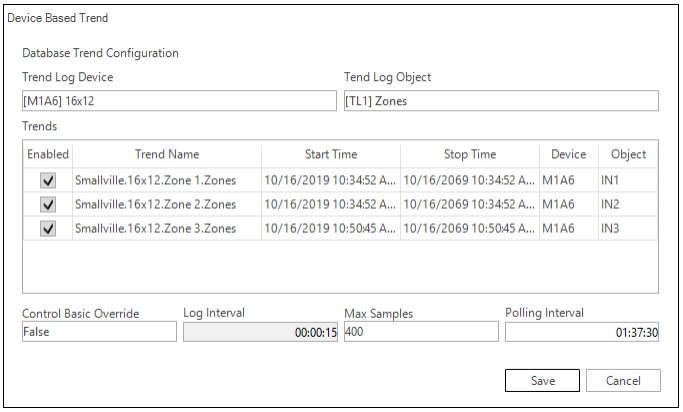
6 Click **Save** when finished.

7 Drag the trend object from the **Device Trend Configuration** 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 KMD device based trend. The dialog opens automatically when a BACnet object is dropped into the Database Trend Configurations list.

Illustration 15–2 KMD Device Trend dialog



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

1. Right-click a device 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.

Properties of the Device Based Trend dialog

Enabled Select the **Enabled** 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 a name for the point trend log as it should appear in the trend group.

Start Time and Stop Time Set the beginning and ending period of the trend log. By default, the beginning date is the day the trend log is added to the Database Trend Configuration list; the end date is the start date with 5 years added.

Control Basic Override When True, the trend log is collecting data under command from Control Basic instead of performing automatic sampling based on the period set by Log Interval. See the procedure *On-demand logging* on page 122.

Log Interval Displays time interval between readings in the Trend Log point in the controller. See the topic *Trend Log points* on page 615 for additional information.

Max Samples Displays the number of samples in the buffer. See the topic *Trend Log points* on page 615 for additional information.

Polling Interval The polling interval must be short enough to collect controller trend before new data replaces old data. The default value is calculated from the maximum number of samples in a controller and the interval in the KMD trend log.

Polling Interval = (Max Samples-10) * Log Interval

Related topics

- *About trend logs* on page 99
- *Configuring KMD controller Trend Logs* on page 121
- *Adding KMD trend logs to a trend group* on page 126

Configuring KMD PC trend logs

A PC trend log is a polling process performed by the KMC Connect trend service. For a detailed overview on PC trends, see *About trend logs* on page 99.

- Any KMD point except trend logs may be selected for a PC trend.
- A point must be added to a trend group to be visible in the trend group.
- A point may be placed in more than one trend group.

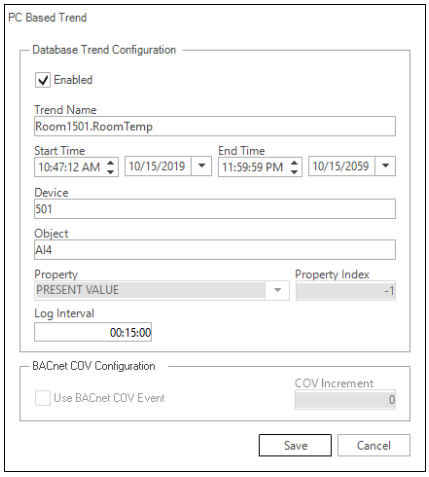
To set up a PC trend log for a KMD point, do the following:

1. On the ribbon, choose **Manager** from the **Trends** group.
2. In the Network Manager list locate the point to trend.
3. Drag the point from the Network Manager list to the **Device Trend Configurations** list in the Trend Manager. The KMD **PC Based Trends** dialog opens. .
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

Edit the setting for PC trends from KMD controllers with the PC Based Trend dialog.

Illustration 15–3 KMD PC Based Trends dialog



To open the dialog:

1. Right-click a PC trend log icon in either the **Device Trend Configuration** or **Trend Groups** list.

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

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

Trend Name Enter the name of the trend as you want it to appear on the trend group. The default value for **Trend Name** is the point name.

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

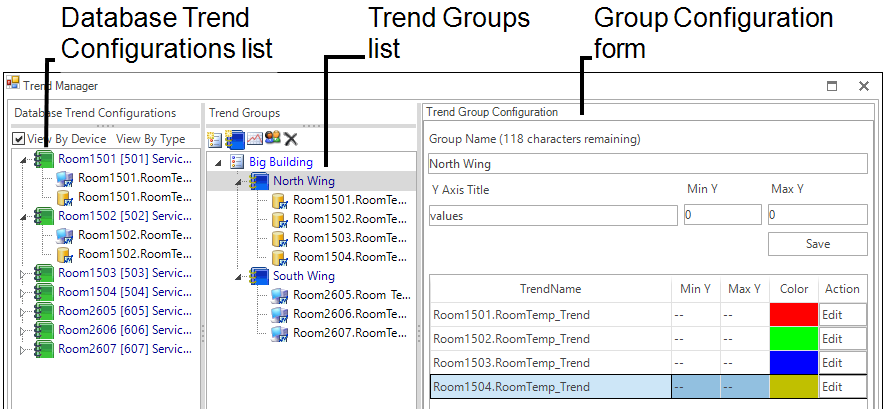
Log Interval Enter the frequency to sample data in **Log Interval**. See *Changing the Network Manager options* on page 48 to change the default value.

Adding KMD trend logs to a trend group

Trend logs are organized in the Trend Selector list with trend categories and groups.

- Device trends use less network bandwidth than PC trends and are the preferred method to add trend logs to the trend group.
- Only PC trend logs and device trend logs placed in a trend group are visible in the trend group.
- Trend groups can be placed only in trend categories.

Illustration 15–4 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

Trend groups can be placed only in trend categories.

1. If there is no trend category in the **Trend Groups** list, 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 group.
   - In the **Y Title** text box, enter a label for the value (y) axis chart that will display in the trend group.
   - 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 Manager 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 list 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.

Related topics

- *About trend logs* on page 99
- *Configuring KMD controller Trend Logs* on page 121
- *Configuring KMD device trend logs* on page 123

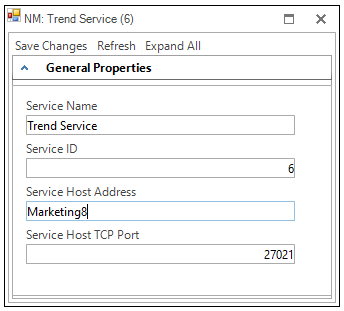
Section 16: Configuring the Trend Service

The KMC Connect Trend Service is the link between a BACnet building automation system and other KMC Connect services.

To gather historical trend data from a KMC Connect managed building automation system, a Trend Service must be installed and running on the computer hosting the protocol driver service. Typically, a Trend Service is installed and configured at the time KMC Connect building services are installed.

- *Configuring an existing service* on page 46
- *Removing a service* on page 47

Illustration 16–1 Trend Service dialog



To configure the Trend Service, do the following:

1. Locate the Trend Service icon in the Network Manager list.
2. Right-click the 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.

Trend Service properties

Service Name A descriptive label of the service. **Service Name** is the name that identifies the service in the Network Manager list. The set of characters used in **Service Name** is restricted to printable characters.

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

Service Host Address This is the address of the computer on which the SQL service is running. The preferred entry is the name of the computer; the IP address may also be used.

Service Host TCP Port This port is used by KMC Connect building services.

KMC Connect

Part IV: Alarms, notifications, and events



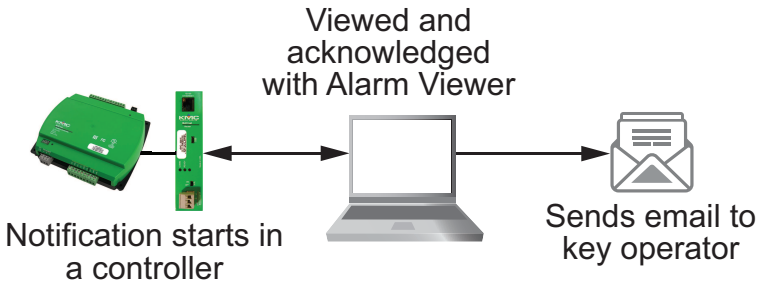
Section 17: An overview of notifications, alarms, and events

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 also involves 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 17–1 KMC Connect alarm process



Origination of notifications Programs within devices monitor conditions. When a value deviates from a normal parameter, a notification is sent 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 BACnet controllers for alarms* on page 135.
- For KMD controllers, see *Configuring KMD controllers for alarms* on page 151.

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 email.

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

Related topics

- *Viewing, acknowledging, and archiving alarms and events* on page 163
- *Emailing operator alarms* on page 171

Section 18: Configuring BACnet controllers for alarms

Topics in this section provide an overview of intrinsic and algorithmic BACnet notifications as well as 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 155.

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 139
- *Configuring BACnet devices for intrinsic alarms* on page 143
- *About algorithmic reporting in BACnet devices* on page 144
- *Configuring event enrollment objects* on page 146
- *Configuring a notification class object* on page 148
- *Change Of Value (COV) notifications* on page 155

Tip: The Alarm wizard is a quick method to set up intrinsic or algorithmic alarms. See the topic *Using the Alarm Wizard for BACnet alarms* on page 136 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.

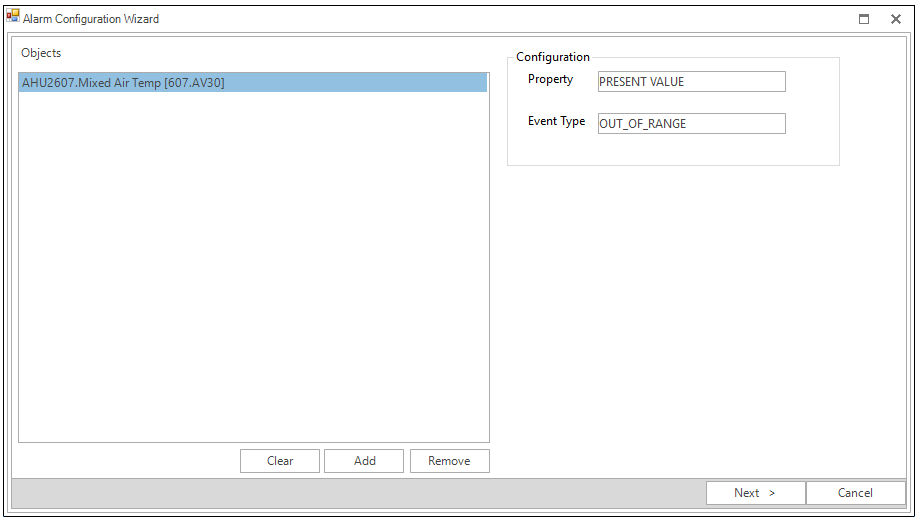
Start the wizard

1 On the ribbon, choose **Configuration Wizard** from the **Alarms** group.



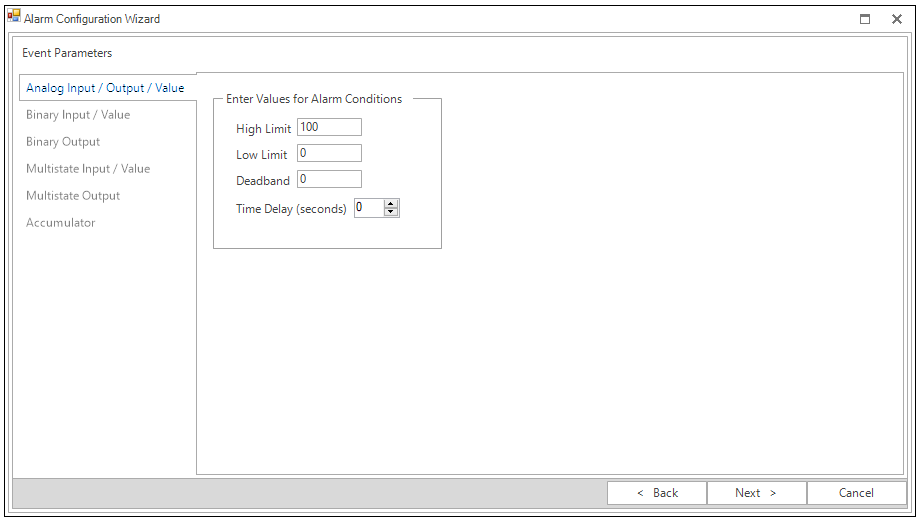
2 Add objects to the Object Selection list.

- 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 138.
- All of the objects must be of the same type. For example all analog input objects or all binary value objects.



3 Click **Next** to advance.

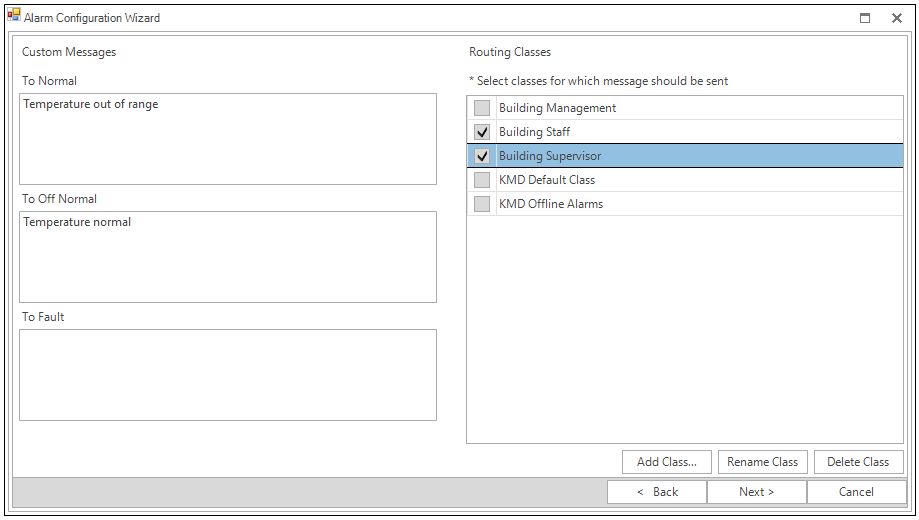
4 Enter the alarm parameters. The exact properties will be depend upon the type of object.



5 Click **Next** to advance.

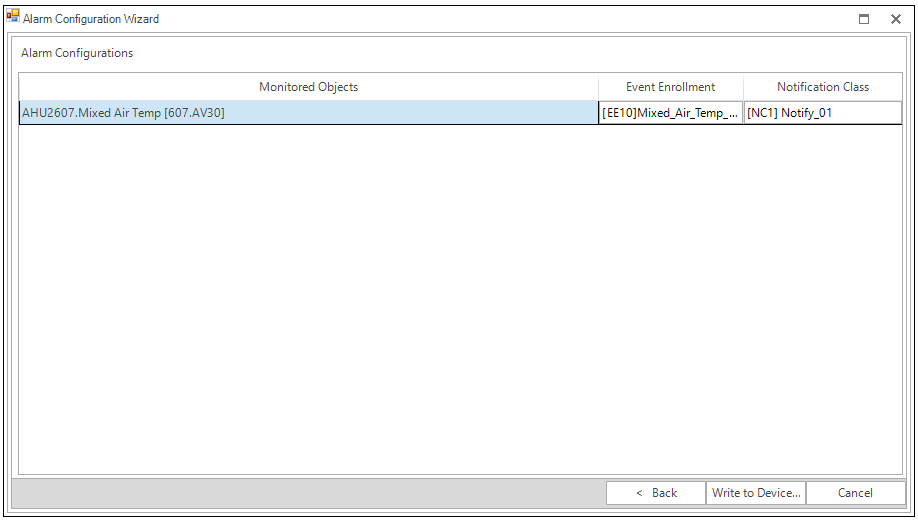
6 Enter custom messages and select the operators that will receive the alarm routing.

- 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 171.



7 Click **Next** to advance.

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



9 To finish, click **Write to Devices**. KMC Connect updates the alarm properties in the selected objects.

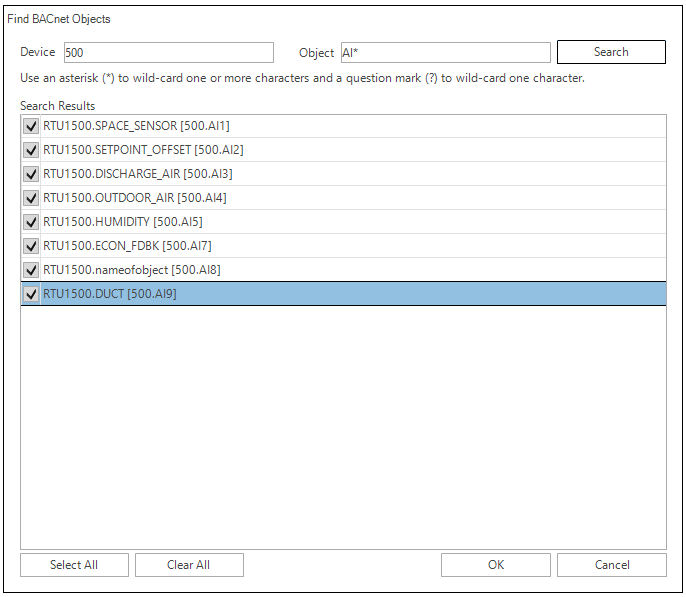
Add by searching

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

To add objects by searching, do the following:

1 Start the Alarm wizard.

2 In the Object Select list, click **Add**. The Find BACnet Objects Dialog opens.



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 171
- *About intrinsic BACnet alarms and events* on page 139
- *About algorithmic reporting in BACnet devices* on page 144
- *Reference to BACnet objects* on page 435

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. A notification can be classified as either an alarm or an event based on the significance of the condition generating the notification.

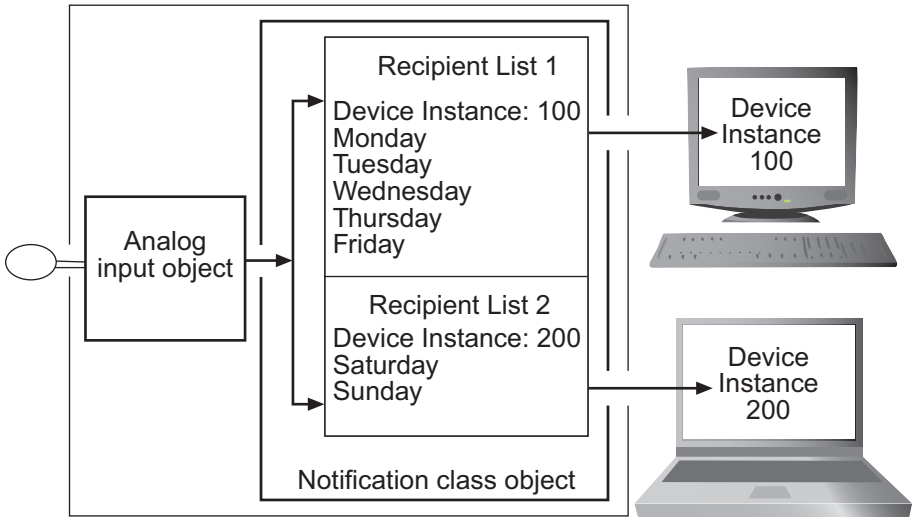
Notification takes place when a monitored value moves outside of predefined limits. The object then sends a notification to a notification class object. Within notification class objects are one or more recipient lists that include the designated notification handling device such as KMC Connect.

An example of intrinsic notification

In the example shown in the diagram *The intrinsic alarm and event notification flow* on page 140, a thermistor is connected to an analog input object. The object is configured to generate notifications 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 notification to the designated notification class object for further handling.
- On weekdays, Recipient List 1 directs the notification to the computer with a workstation configured as device instance #100.
- On weekends, Recipient List 2 sends the notification to the computer with a workstation configured as device instance #200.
- The operator workstation processes the notification. 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 18–1 The intrinsic alarm and event notification flow



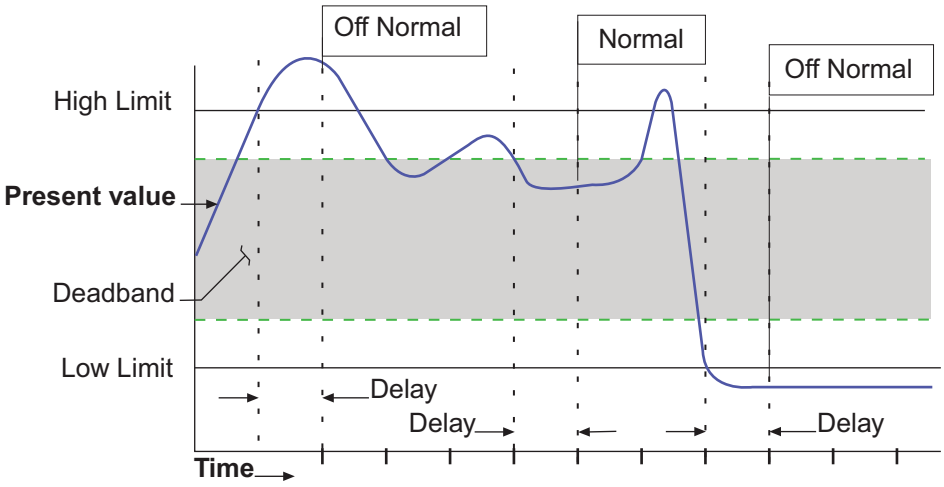
Notifications in analog objects

Each BACnet object that supports notifications contains properties that define the conditions that initiate a notification.

- 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 the present value falls outside of predefined limits.

Illustration 18–2 Example of Off Normal and Normal events in analog objects



All of the following conditions are required to generate 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*.
- *Low/High* must be selected in *Limit Enable*.
- The selection in *Event Enable* must include *Off Normal*.

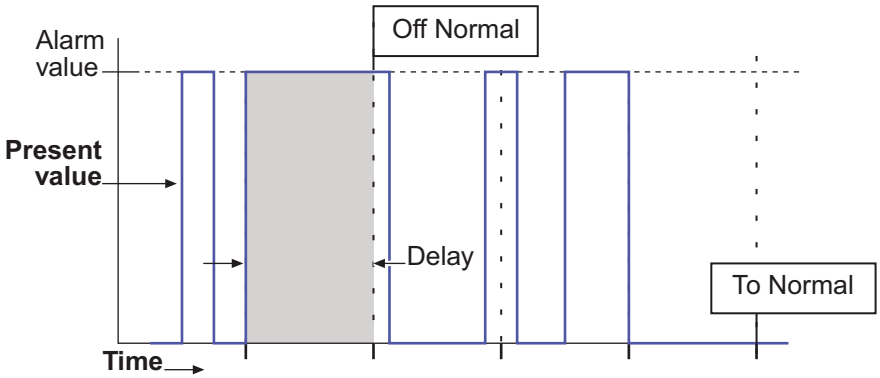
All of the following conditions are required to generate a *To Normal* event when *Present Value* returns to a normal value:

- *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*.
- *Low/High* must be selected in *Limit Enable.*
- 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 a binary object configured to generate notifications when the present value falls outside of predefined limits.

Illustration 18–3 Example of Off Normal and To Normal notifications



All of the following conditions are required to generate *To Off Normal* notifications in a binary object.

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

All of the following conditions are required to generate a *To Normal* notification in binary objects:

- *Present Value* must remain unequal to the value specified by *Alarm Value* for the period specified by *Time Delay*.
- 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. 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 143.
   - For loop objects or binary input, output or value objects, see *Notifications in binary objects* on page 144.
2. Configure a notification class object to send the notification to a handler. See *Configuring a notification class object* on page 148.



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 136.

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 archiving alarms and events* on page 163.
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 148.

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 archiving alarms and events* on page 163.
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 148.

Related topics

- *About intrinsic BACnet alarms and events* on page 139
- *Notification objects* on page 490

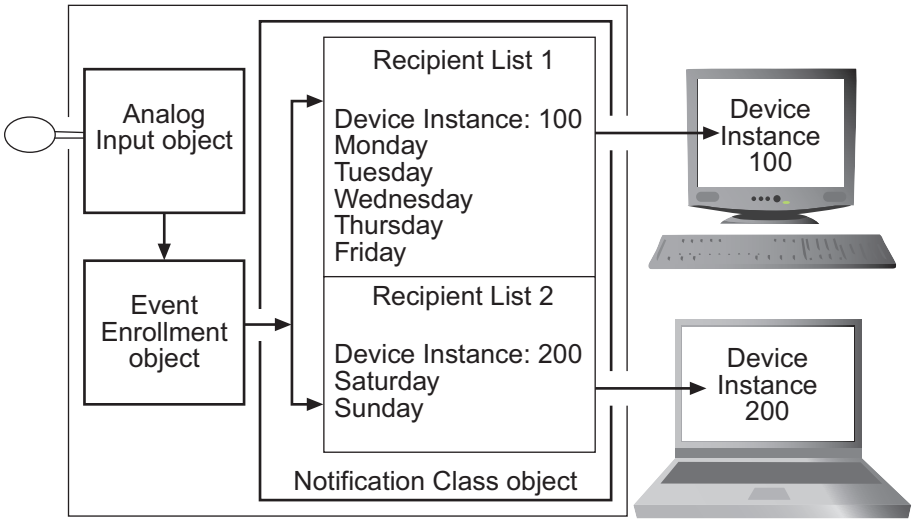
About algorithmic reporting in BACnet devices

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 145, 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 18–4 Algorithmic notification flow



Related topics

- *Configuring event enrollment objects* on page 146
- *Event enrollment objects* on page 451
- *Notification objects* on page 490
- *About intrinsic BACnet alarms and events* on page 139

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 146.
   - For loop objects or binary input, output or value objects, see *Configuring an event enrollment object for binary values* on page 147.
2. Next, configure the notification class object to send the notification to a handler. See *Configuring a notification class object* on page 148.

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 136.

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 140.

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/AlarmProperties**.

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 archiving alarms and events* on page 163.

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 148.

Configuring an event enrollment object for binary values

The following procedure describes configuring an event enrollment 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 142.

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.
   - The object to monitor
   - The binary property to monitor

7 Expand the section **Event/AlarmProperties**.

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 archiving alarms and events* on page 163.

11 Expand the section **Alarm Routing Properties**.

12 Select the **Class Routing** from the list.

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

14 Configure the notification class object referenced in step 9. See *Configuring a notification class object* on page 148.

Related topics

- *About algorithmic reporting in BACnet devices* on page 144
- *Event enrollment objects* on page 451
- *Notification objects* on page 490
- *Viewing all event enrollment subscriptions* on page 160

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 objects* on page 490
- *Configuring BACnet devices for intrinsic alarms* on page 143
- *Configuring event enrollment objects* on page 146
- *Configuring email notifications* on page 172

Section 19: Configuring KMD controllers for alarms

Topics in this section describe the process of creating alarm notifications within KMD series controllers.

KMD series controllers initiate alarm notifications within Control Basic programs.

- *Control Basic alarm statements* on page 151
- *Alarm classes for KMD controllers* on page 153
- *Setting up KMD alarms* on page 154

For BACnet devices and controllers, see the topic *Configuring BACnet controllers for alarms* on page 135.

Control Basic alarm statements

Alarm notifications for KMD controllers are initiated with either of the two Control Basic alarm statements.

- *ALARM* on page 252
- *DALARM* on page 268

When one of the alarm statements initiates an alarm notification, the notification is held in a Tier 1 controller until KMC Connect retrieves and processes it. The Tier 1 controller holding the notification is either the controller in which the notification was created or the Tier 1 controller to which the Tier 2 network is connected.

ALARM statement

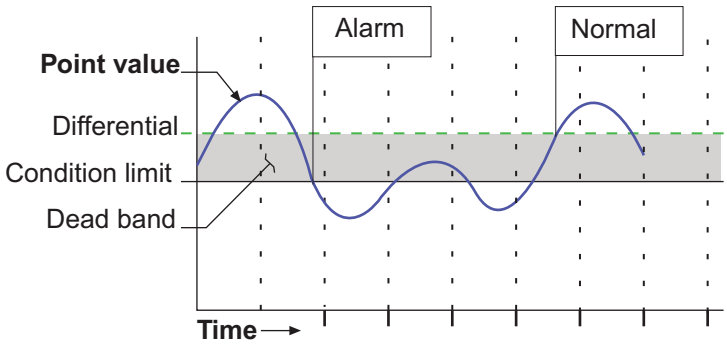
The ALARM Control Basic statement initiates a notification based on the value of a point in a controller. If the value of the point exceeds the limit, an alarm notification is sent to the network. The alarm status for that point does not return to normal until the value of the point is outside of the alarm dead band. The dead band is set by a combination of the condition limit and a differential value.

The following Control Basic statement initiates an alarm when Input Point IN1 drops below 34 degrees. The alarms status returns to normal after the input rises above 37 degrees.

```
ALARM IN1 < 34 , 3 , Air temperature is close to
freezing.
```

The diagram *Example alarm statement* on page 152 is a diagram of the alarm process.

Illustration 19–1 Example alarm statement



DALARM statement

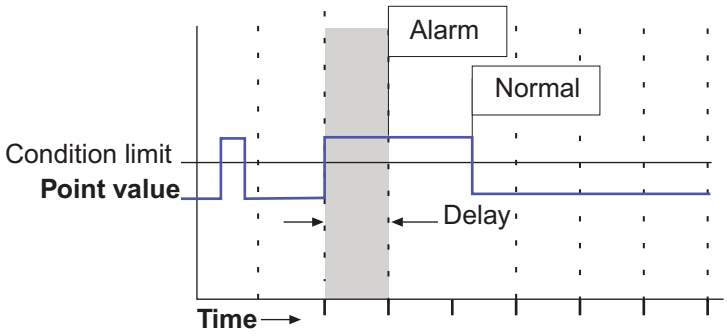
The *DALARM* Control Basic statement initiates a notification only after the point is outside of the limit for a specific period. This prevents multiple alarms from being issued when the state exceeds the conditions for a brief period that can be ignored. DALARM is typically the best alarm statement to use for digital points.

The following Control Basic statement initiates an alarm when the value at input IN1 is *True* (*1*) for longer than 5 seconds. As soon as the point returns to *False* (*0*), the state of the alarm changes to normal.

```
DALARM IN1 <> False , 5 , Door is open!
```

The diagram *Example delay alarm statement* on page 152 shows how delay works in an alarm statement.

Illustration 19–2 Example delay alarm statement



Related topics

- *An overview of notifications, alarms, and events* on page 133
- *Viewing, acknowledging, and archiving alarms and events* on page 163
- *Setting up KMD alarms* on page 154
- *Alarm classes for KMD controllers* on page 153
- The Control Basic statements *ALARM* on page 252 and *DALARM* on page 268

Alarm classes for KMD controllers

There are three unique alarm classes for KMD alarms.

- KMD Default Class
- KMD Offline Alarm
- WinControl email notifications

KMD Default Class This is a default class for the KMD protocol. The KMD Default Class receives notifications that originate automatically within a KMD controller. The KMD PDS periodically polls all Tier 1 controllers for these notifications. Notifications include the following conditions:

- A Control Basic program did not run correctly.
- A Tier 2 device is offline.
- A Tier 1 device is offline.
- BACnet issues in BACnet licensed Tier 1 controllers.
- Changes to Ethernet connections.
- Internal memory problems.

The alarms generated for KMD Default Class are listed in the topic *KMD Tier 1 alarm messages* on page 625.

KMD Offline Alarm Class This is a default alarm class for the KMD protocol. The KMD PDS periodically polls any attached KMD networks for the presence of KMD Tier 1 controllers. If a previously discovered controller is not online, its status is changed in the Network Manager list. All Tier 2 controllers connected to the Tier 1 controller are then also set to offline. There are four notifications generated by the KMD PDS for this alarm class:

- KMD Network Offline
- KMD Network Online
- KMD Device Offline
- KMD Device Online

WinControl email alarms KMC Connect receives and processes WinControl email alarms. Email alarm classes respond to the E1-E8 parameter in the Control Basic statements *ALARM* on page 252 and *DALARM* on page 268. These email classes have to be added to the Classes Of Alarms list in the Alarm Routing Manager tool.

- The alarm classes are *E1* to *E8*.
- They do not require set up other than operator notification.
- The alarm classes E1 to E8 may be bound to other incoming notifications.

Related topics

- *An overview of notifications, alarms, and events* on page 133
- *Viewing, acknowledging, and archiving alarms and events* on page 163

- *Control Basic alarm statements* on page 151
- *Control Basic alarm statements* on page 151, *ALARM* on page 252 and *DALARM* on page 268

Setting up KMD alarms

Configuring an alarm for a KMD point is a two-part process.

1. Write a Control Basic statement to initiate the alarm notification for a monitored point. The Control Basic statement is added to one of the Control Basic programs within the controller that contains the monitored point.
2. Select the Alarm Class Routing for the point monitored for alarm.

To set up an alarm in a KMD controller, do the following.

1. Locate the controller in the Network Manager list.
2. Open the Control Basic program in which the alarm statement will run. Alarms are best added to a higher numbered program, preferably the highest numbered program within the controller.
3. Add a program line that includes either an ALARM or DALARM statement.
4. Save and close the program.
5. Locate and open the monitored point in the Network Manager list.
6. Expand the section **Alarm Routing Properties** and then select the **Class Routing** from the list.
7. When done, click **Save Changes** at the top of the object tab.

Related topics

- *An overview of notifications, alarms, and events* on page 133
- *Viewing, acknowledging, and archiving alarms and events* on page 163
- *Alarm classes for KMD controllers* on page 153
- *Control Basic alarm statements* on page 151
- Control Basic keywords *ALARM* on page 252 and *DALARM* on page 268

Section 20: 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 155
- *Subscribing to an object on a configuration page* on page 157
- *Subscribing to a property on a configuration page* on page 158
- *Subscribing trend logs to a COV server device* on page 160
- *Unsubscribing from COV subscriptions* on page 159

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**— Identifies the monitored BACnet object.
- **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 an operator 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 **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 155
- *Subscribing to a property on a configuration page* on page 158
- *Unsubscribing from COV subscriptions* on page 159

Subscribing to a property on a configuration page

Open configuration pages for BACnet objects may receive COV notifications from 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 does subscribing to an object.

Use one of 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**.

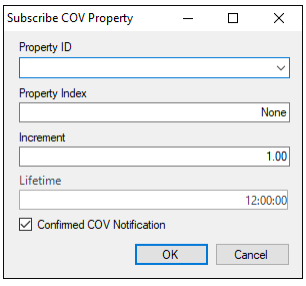
From the object configuration tab

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 20–1 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 is necessary to maintain the subscription, 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 the **Present Value** text box 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 and choose one of the following options:

- 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 110.

Viewing all event enrollment subscriptions

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

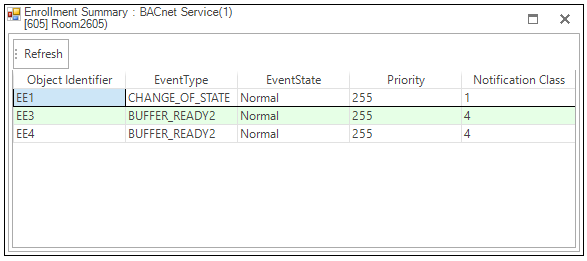
1 Locate the device in the Network Manager list.

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 20–2 Summary of event enrollment subscriptions



Related topics

- *Change Of Value (COV) notifications* on page 155
- *Event enrollment objects* on page 451
- *Configuring event enrollment objects* on page 146

Section 21: Viewing, acknowledging, and archiving 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 133
- *Viewing alarms* on page 163
- *Acknowledging alarms* on page 165
- *Archiving alarms in KMC Connect* on page 166
- *Filtering alarms* on page 168
- *Disabling alarms by network* on page 169
- *Viewing a summary of BACnet notifications* on page 166
- *Using the Alarm Monitor bar* on page 167

Viewing alarms

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

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

The Alarm Viewer tab opens a list of points and objects that triggered 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 164.
- The icons in the second column indicate the status of the notification. See the table *Acknowledgment status icons* on page 164.
- 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 21–1 The Alarm Viewer

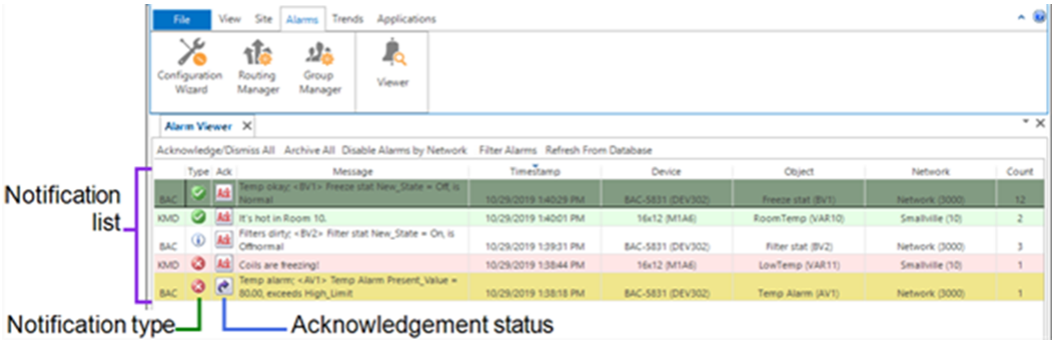


Table 21–1 Notification type icons

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 21–2 Acknowledgment status icons

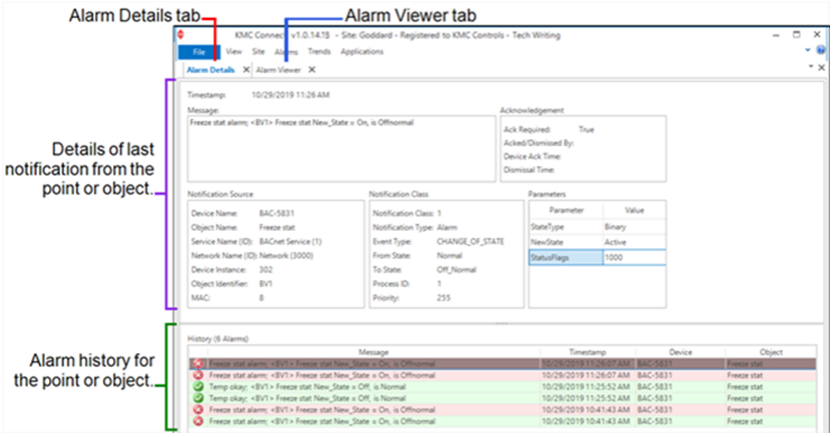
Icon	Description
	A notification in the database is waiting for an acknowledgment from an operator.
	A notification that does not require an operator acknowledgment.
	KMC Connect is updating the status.
	An operator has acknowledged the notification.

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 21–2 Alarm Details tab



Related topics

- *An overview of notifications, alarms, and events* on page 133
- *Acknowledging alarms* on page 165
- *Archiving alarms in KMC Connect* on page 166
- *Using the Alarm Monitor bar* on page 167
- *Filtering alarms* on page 168
- *Disabling alarms by network* on page 169

Acknowledging alarms

Notifications that require an operator acknowledgment are identified with the Ack icon in the **Ack** column of the notification list. Alarms that are acknowledged remain in the notification list of the Alarm Viewer.

Do the following to acknowledge an alarm in KMC Connect:

1 On the ribbon, choose **Viewer** from the **Alarms** group.



2 Do one of the following:

- To acknowledge a single alarm, click the **Ack** icon in the alarm status column of the notification list.
- To acknowledge all alarms, click **Acknowledge/Dismiss All** at the top of the tab.

Related topics

- *An overview of notifications, alarms, and events* on page 133

- *Viewing alarms* on page 163
- *Archiving alarms in KMC Connect* on page 166
- *Using the Alarm Monitor bar* on page 167
- *Filtering alarms* on page 168
- *Disabling alarms by network* on page 169

Archiving alarms in KMC Connect

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

1 On the ribbon, choose **Viewer** from the **Alarms** group.



2 Do one of the following:

- To archive a single alarm, right-click an alarm in the **Alarm Viewer** tab and choose **Archive** from the shortcut menu.
- To archive all alarms, click **Archive All** at the top of the tab.

Related topics

- *An overview of notifications, alarms, and events* on page 133
- *Viewing alarms* on page 163
- *Acknowledging alarms* on page 165
- *Using the Alarm Monitor bar* on page 167
- *Filtering alarms* on page 168
- *Disabling alarms by network* on page 169

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

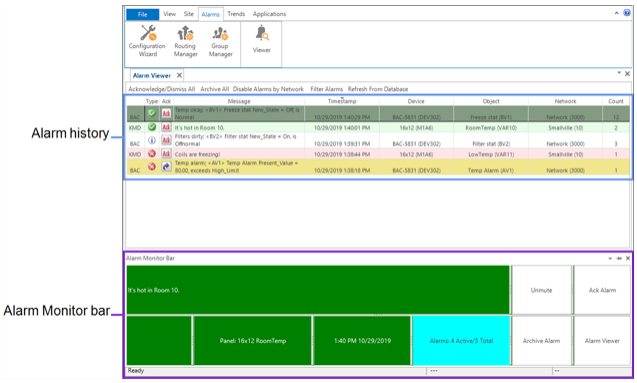
- *An overview of notifications, alarms, and events* on page 133
- *Viewing alarms* on page 163
- *Acknowledging alarms* on page 165
- *Archiving alarms in KMC Connect* on page 166
- *Using the Alarm Monitor bar* on page 167
- *Filtering alarms* on page 168
- *Disabling alarms by network* on page 169

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. To change when the Alarm Monitor Bar opens, see the topic *Options for KMC Connect* on page 38.

The Alarm Monitor bar can also be opened from the ribbon by choosing **Alarm Monitor Bar** from the **View** group.

Illustration 21–3 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 color codes.

Table 21–3 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 163.

Filtering alarms

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

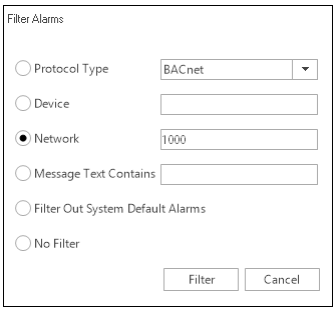
- Protocol type
- Device instance number
- Network number
- Contents of the message text
- System default alarms.

To filter alarms, do the following:

1 At the top of the Alarm Viewer, click **Filter Alarms**.



2 In the Filter Alarms dialog, choose the criteria for filtering.



3 Click **Filter** when finished.

Related topics

- *An overview of notifications, alarms, and events* on page 133
- *Viewing alarms* on page 163
- *Acknowledging alarms* on page 165
- *Archiving alarms in KMC Connect* on page 166
- *Disabling alarms by network* on page 169
- *Viewing a summary of BACnet notifications* on page 166
- *Using the Alarm Monitor bar* on page 167

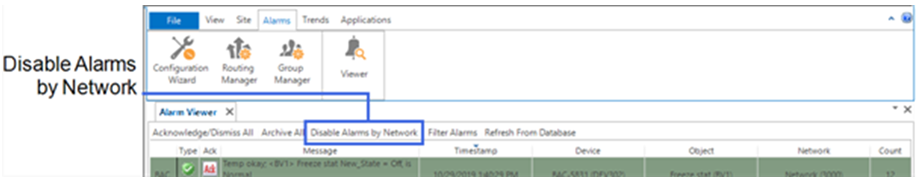
Disabling alarms by network

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

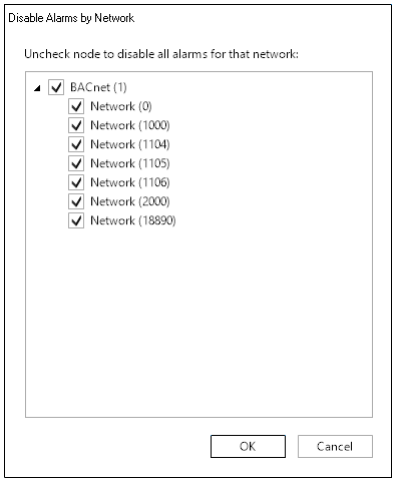
Alarms from disabled networks are not handled or stored by KMC Connect. They cannot be retrieved at a later date.

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.

Related topics

- *An overview of notifications, alarms, and events* on page 133
- *Viewing alarms* on page 163
- *Acknowledging alarms* on page 165
- *Archiving alarms in KMC Connect* on page 166
- *Filtering alarms* on page 168
- *Viewing a summary of BACnet notifications* on page 166
- *Using the Alarm Monitor bar* on page 167

Section 22: 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 adding alarms and events to the alarm viewer list, alarms and events can be routed to specific users by email.

- *Configuring email notifications* on page 172
- *Filtering alarms by security groups* on page 175
- *Configuring the Notification Service* on page 177

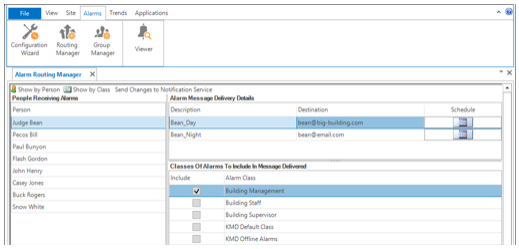
Setting up alarm emails

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 133.
- To set up operator notifications see *Configuring email notifications* on page 172.

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

Illustration 22–1 Alarm Routing Manager



The components of the Alarm Routing Manager

People Receiving Alarms This is a list of operators that will receive alarms and event messages by email.

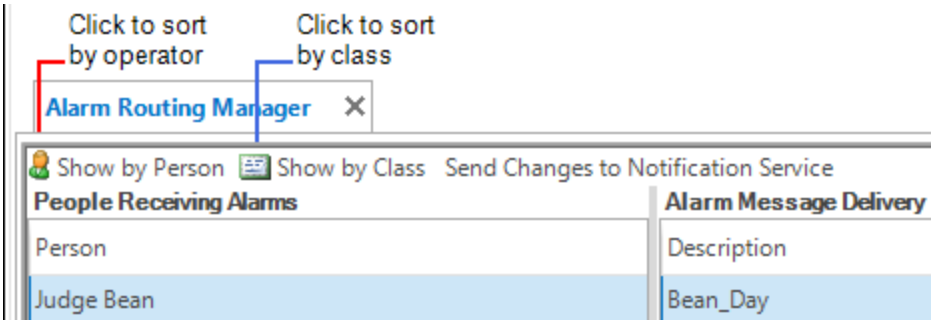
Alarm Message Delivery Details This property sets the schedule and email address or addresses for each person.

Classes Of Alarms To Include in Message Delivered An Alarm Class is a group of operators that need to receive an alarm notification in email messages. When an alarm or event is received by the KMC Connect Notification Service for an alarm class, email messages are sent to recipient operators based on the configuration in Alarm Routing Manager. The alarm or event is assigned to an alarm class in the device that initiates the alarm or event.

Changing the listing view

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

Illustration 22–2 Changing the display listing



Related topics

- *Configuring BACnet controllers for alarms* on page 135
- *Configuring KMD controllers for alarms* on page 151
- *Configuring email notifications* on page 172
- *Viewing, acknowledging, and archiving alarms and events* on page 163

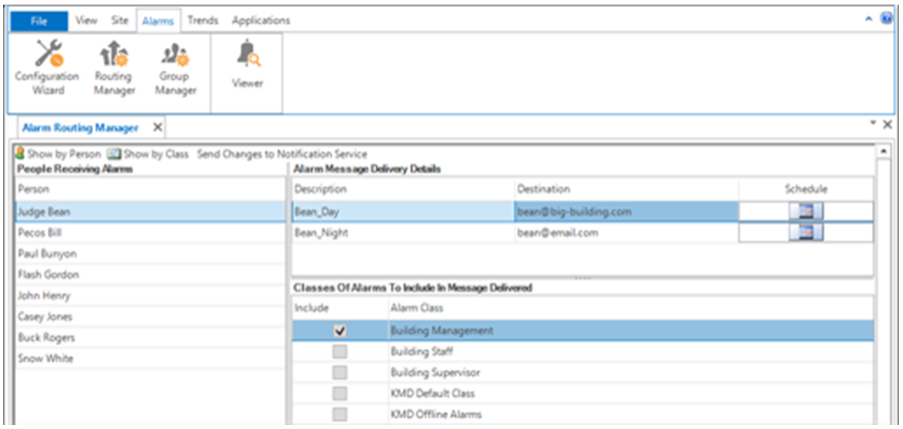
Configuring email notifications

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.

The notification process is explained in the topic *An overview of notifications, alarms, and events* on page 133.

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

Illustration 22–3 Alarm Routing Manager



Setting up emails for alarms and events

Setting up email notification is a three-step process.

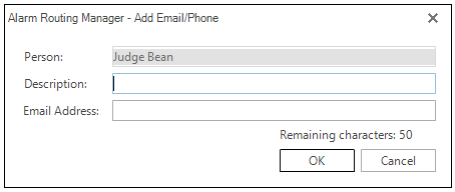
1. Adding operator names.
2. Assigning one or more email addresses to each name.
3. Assigning names to one or more Alarm Classes.

To add a name, do the following:

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

To assign an email address to a person, do the following:

1. Choose a name from the **Person** list.
2. Under the **Alarm Message Delivery Details** list, right-click and then select **Add Email** from the shortcut menu.



3. In the **Add Email/Phone** dialog, do the following:
   a. Enter the email address.
   b. Enter a description for the delivery detail.
   c. Click **OK** when the entry is complete.
4. Click the Schedule button and select the times of day for the entry.
5. Click **Close** when finished.

To assign a person to one or more alarm classes, do the following:

1. Choose a name from the **Person** 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: The alarm class must also be enabled in every object or point that originates an event for which this operator must be notified. See *Configuring BACnet controllers for alarms* on page 135 for details about configuring devices and objects for event notification.

To add a new alarm class, do the following:

1. Right-click under the **Alarm Class** list, and then choose **Add Class** from the shortcut menu.
2. Enter the new **Class Name**.
3. Click **OK** when finished.

The components of the Alarm Routing Manager

People Receiving Alarms This is a list of operators that will receive alarms and event messages by email.

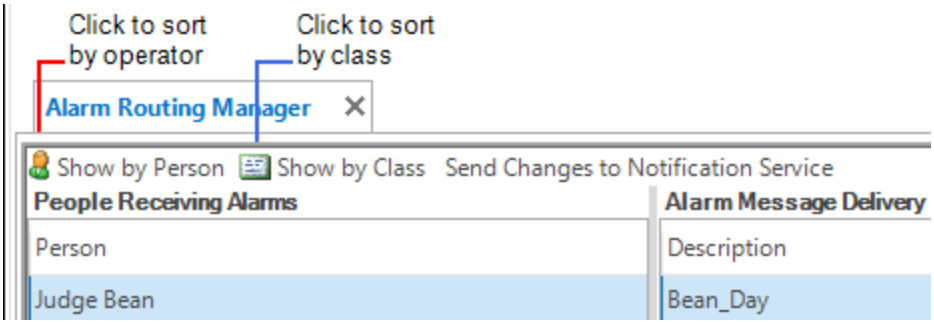
Alarm Message Delivery Details This property sets the schedule and email address or addresses for each person.

Classes Of Alarms To Include in Message Delivered An Alarm Class is a group of operators that need to receive an alarm notification in email messages. When an alarm or event is received by the KMC Connect Notification Service for an alarm class, email messages are sent to recipient operators based on the configuration in the Alarm Routing Manager. The alarm or event is assigned to an alarm class in the device that initiates the alarm or event.

Changing the listing view

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

Illustration 22–4 Changing the display listing



Section 23: 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 viewing, 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 77.

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 .

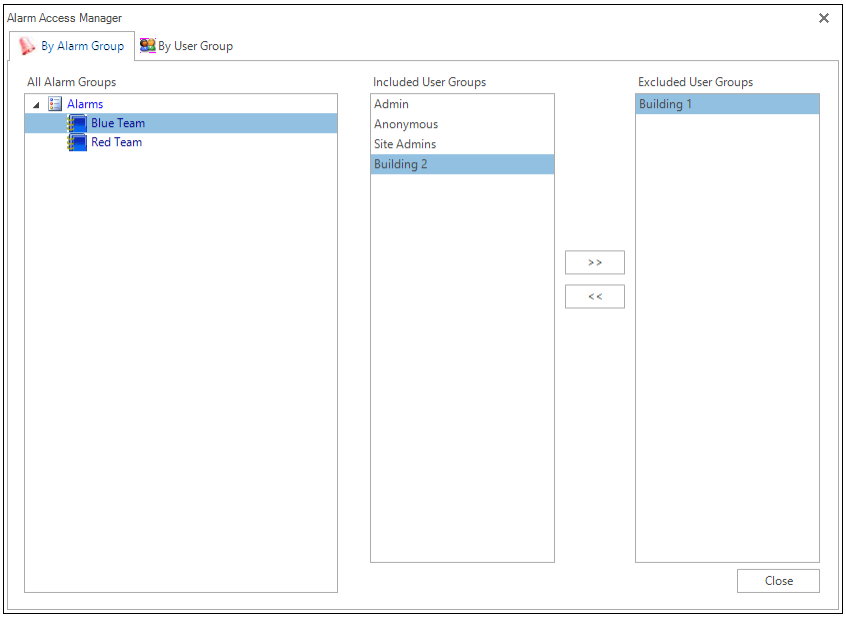
5 In the Alarm Group Manager list, right-click a group icon and choose **Alarm Access Manager** from the shortcut menu. The Alarm Access Manager dialog screen opens.

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

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

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

Illustration 23–1 Alarm Access Manager dialog screen



Section 24: Configuring the Notification Service

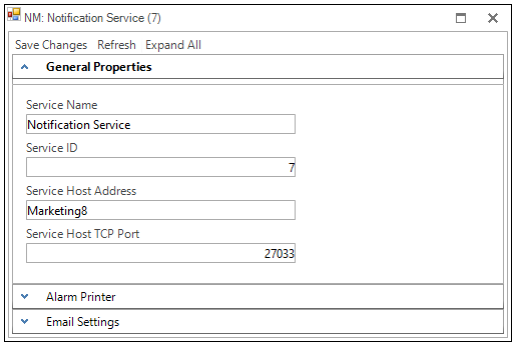
The KMC Connect Notification Service is the link between a BACnet building automation system and KMC Connect.

To process alarms, events and notifications originating in a KMC Connect managed building automation system, a Notification Service is installed as part of the KMC Connect installation. Setting the following Notification Service properties are covered in this topic.

- *General Properties* on page 177
- *Alarm Printer* on page 178
- *Email settings* on page 178

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 icon.
4. From the shortcut menu, choose **Configure Service**. A configuration tab opens in the workspace.



5. Expand one of the following areas:
   - *General Properties*
   - *Alarm Printer*
   - *Email settings*
6. Make the changes to the service.
7. When changes are complete, click **Save Changes** at the top of the 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

The account credentials supplied by the server administrator include the following:

Outgoing Mail Settings

Server Type The name of the outgoing mail server.

Server Address The address of the outgoing mail server. The address may take the form of a URL (e.g., bigbuilding.com) or an IP address (e.g., 192.0.2.0).

Secure Select this box if messages requires a secure connection. When Secure is checked, the software will set the EnableSsl property to true, and the email server will determine TLS or SSL according to which type is supported.



Note: Email service providers may require setup and configuration of third-party app passwords for proper operation. Refer to your email service provider's instructions for the steps required to support third-party apps.

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

Email A user account name (e.g., TCalarms).

Password This is a third-party application code generated by the email service provider. Refer to your email service provider's instructions for generating and managing third-party application passwords.

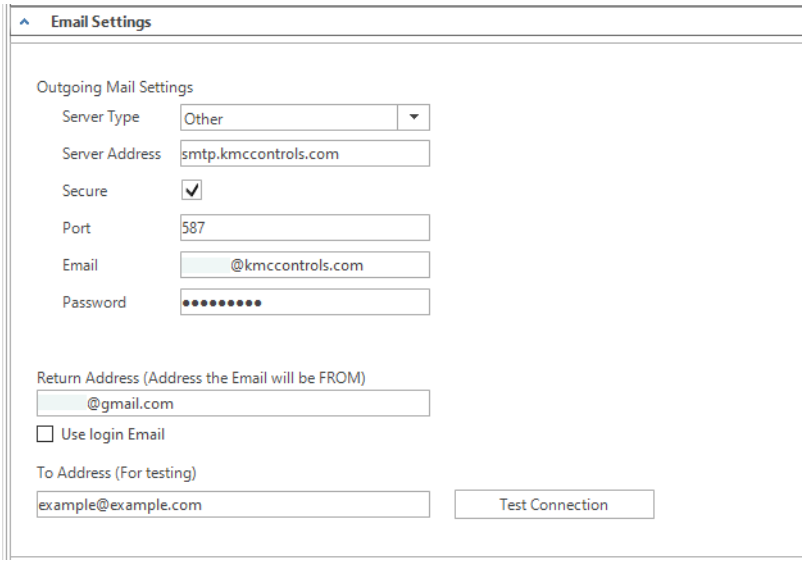
Return Address (Address the Email will be FROM) The reply address that is included as part of the message (e.g., maintenance@bigbuilding.com).

Use login Email Select this box if the return address will be the same as the log-in email address.

To Address (For testing) Type the recipient address (e.g., superTech@bigbuilding.com) for an outgoing test message. This address is used only to test the connection to the email server.

Test Connection Click **Test Connection** to send a test message to the email recipient in **To Address (For testing)**.

Illustration 24–1 Email Settings tab

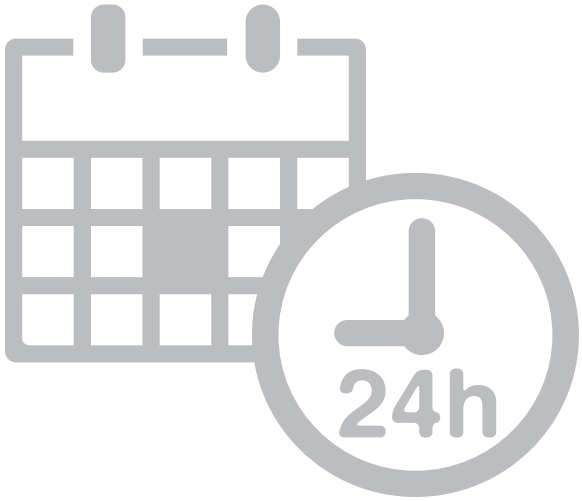


Related topics

- *Configuring an existing service* on page 46
- *Removing a service* on page 47
- *Opening and closing the Network Manager pane* on page 42

KMC Connect

Part V: Schedule management



Section 25: About BACnet schedules

In a KMC Connect managed building automation system, BACnet schedules are a convenient method for controlling recurring or future events without extensive programming. Topics in this section describe the Schedule and Calender objects and cover setting up schedules with KMC Connect.

BACnet controllers contain three types of schedules that are maintained within two types of BACnet objects.

- Weekly schedules are a list of daily events that are maintained within Schedule objects.
- Exception schedules override special days in the Weekly schedule and are maintained also within Schedule objects.
- Calendar objects maintain common Exception schedules that can be shared among Schedule objects.

In a Schedule object, the Present Value property changes state depending on the time-of-day and the day-of-the-week in the controller that maintains the object. The schedule for each day is a list of time-value pairs. Each time-value pair includes a time-of-day and a state or value. As the time in the controller matches the time and day-of-the-week in a time-value pair, the Present Value of the object changes to the state or value defined by the time-value pair. The state or value of objects referenced by the Schedule object also change state or value.

Unless the current day is an Exception day, Schedule objects follow the Monday-Sunday daily schedules in the Weekly Schedules area. Exception days take place when the date in the controller's Exception schedule matches the date in the controller. Then, the state of the Weekly schedule is controlled by the time-value pairs in the Exception schedule. Days for the Exception schedule can be designated from within the Schedule object or a Calendar object.

In a Calendar object, the Present Value property changes state for the entire day. When the calendar maintained in the controller reaches one of the Exception days, the object changes to True, otherwise the Present Value property is False. The entries for Exception days maintained in a Calendar object can be for individual days, the days between two dates, or a day of the week or day of the month. Schedule management is simplified by using Calendar objects because several Schedule objects can use the same Calendar object for common Exception days.

Related topics

- *Configuring BACnet Schedule objects* on page 184
- *Configuring BACnet Calendar objects* on page 186
- *Schedule objects* on page 504
- *Calendar objects* on page 441

Configuring BACnet Schedule objects

BACnet Schedule objects directly control the value of properties of one or more BACnet objects designated in the Object Reference List. The control is based on the times and values entered into the Weekly Schedule property.

To configure a BACnet Schedule object, do the following.

1. Open the Network Manager.
2. Expand the Network Manager list to reveal the controller, the Schedule Objects folder, and the Schedule object icons in the folder.
3. Double-click an icon to open the Schedule object configuration tab.
4. In the **General Properties** area, enter the following:
   - Enter a name and description for the object.
   - In the Present Value area, choose a **Data Type**.
   - Drag one or more BACnet objects to the **Object Reference List**. All objects must be of the same data type and must match Data Type in Present Value.
   - In **Start Date** and **End Date** in **Effective Period**, set the dates to make the schedule active.
   - In the **Schedule Default** area, choose a **Data Type**. Data Type must match Data Type in Present Value.
   - Enter a value in **Schedule Default**.
   - Enter a value in **Priority for Writing**.
5. Expand **Weekly Schedules** and add time-value pairs. See the following procedure *Entering time-value pairs* on page 184.
6. For special days such as holidays, special events or maintenance days, expand **Exceptions Schedule** and add dates and time-value pairs.
7. Click **Save Changes** at the top of the configuration tab.

Entering time-value pairs

Every time-value pair includes a time-of-day and a state or value. When the day and time in a controller matches the day and time in a time-value pair, the Schedule object updates the Present Value property to the value in the time-value pair. It updates also the properties of the objects in the Objects Reference List to the value in the 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. Next to the time value, click the up or down arrows .

To remove a time-value pair

Click the remove button - to delete a time-value pair.

Configuring an Exception schedule

On special days, an Exception schedule overrides the time-value pairs of the Weekly schedule. Typically, the special days are holidays, special events, or other days that change the normal schedule in a Schedule object. For special days that are common to two or more schedule objects, use a Calendar object to designate special days.

To add an exception schedule to a schedule object, do the following:

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 as described in the procedure *Entering time-value pairs* on page 184.

To add a Calendar object to the Exception schedule

For details on the Calendar object see the topic *Calendar objects* on page 441.

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 as described in the procedure *Entering time-value pairs* on page 184.

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**.

Related topics

- *About BACnet schedules* on page 183
- *Configuring BACnet Calendar objects* on page 186
- *Schedule objects* on page 504.

Configuring BACnet Calendar objects

A BACnet Calendar object is a standard BACnet object that contains a list of calendar dates. Typically, the dates are holidays, special events, or other special days that need to be added to the Exception schedule in a Schedule object. Schedule management is made easier with Calendar objects because several Schedule objects can use the same Calendar object for common Exception days.

To configure a Calendar object, do the following:

1. In the Network Manager list, locate the device, the folder of Calendar objects, and the Calendar object  icon.
2. Double-click the object icon to open it.
3. Add or delete dates as needed.
4. For each date entry, add time-value pairs as described in the procedure *Entering time-value pairs* on page 184.
5. When finished, click **Save**.

Adding a calendar entry

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.

Changing an entry

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

- *About BACnet schedules* on page 183
- *Configuring BACnet Schedule objects* on page 184
- *Calendar objects* on page 441

Section 26: About KMD Schedules

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

Every KMDigital controller contains three types of schedules within two different types of schedule points.

- Weekly schedules for recurring daily events are maintained within Weekly Schedule points.
- Override schedules for special days are included also in Weekly Schedule points.
- Schedules for special days of the year are maintained within Annual Schedule points.

Weekly Schedule points change state depending on the time-of-day and the day-of-the-week in the controller that maintains the schedule. As the time in the schedule falls within a time range in a daily schedule, the schedule changes state to match the block it is in. For example, if 08:00 is entered in the first On block, the schedule changes to *Active* when the clock in the controller reaches 0800 hours (8:00 A.M.). Because Weekly Schedule points do not directly control any other points, use Control Basic to set the value of another point or local variable to the state of the schedule.

Unless an override is active, Weekly Schedule points follow the Monday-Sunday daily schedules in the Weekly Schedules area. Overrides are applied by adding a KMDigital point to either the Override Object 1 or Override Object 2 text boxes. Then, when either of the override points change to *True*, the Weekly Schedule follows the schedule in the Weekly Schedule area under Override1 or Override2. The point in Override Object 1 takes precedence. For instance, if Override Object 1 is *Active*, the schedule for Override Object 2 is ignored even if Override Object 2 is also *Active*.

Annual Schedule points change state for the entire day. When the calendar maintained in the controller reaches one of the days designated by the override point, the Annual Schedule point changes to *Active*. Typically, Annual Schedule points designate holidays, maintenance or other special days. Use an Annual Schedule point to override a Weekly Schedule point by adding its mnemonic to either Override Object 1 or Override Object 2 in the Weekly Schedule object.

Related topics

- *Configuring KMD Weekly Schedule points* on page 190
- *Configuring KMD Annual Schedule points* on page 191
- *Weekly Schedule points* on page 618
- *Annual Schedule points* on page 589

Configuring KMD Weekly Schedule points

Use Weekly Schedule points for events that take place at the same time, every day on normal days. A scheduled time in a Weekly Schedule point sets the state of the point to *Active* when the time in the controller matches the scheduled time. Occupancy and lighting are two examples of events that are typically scheduled with Weekly Schedule points.

To enter weekly schedules, do the following:

1. Open the Network Manager.
2. Expand the Network Manager list to reveal the controller, the Weekly Schedules folder, and the Weekly Schedules icons  in the folder.
3. Double-click an icon to open the Weekly Schedule configuration tab.
4. Under **General Properties**, enter the following:
   - Name
   - Description
5. Optional: In Override Object 1 and Override Object 2, enter the points that will control the overrides. Typically, the override points are Annual schedules. Use mnemonics (AS1, AS2, etc.) to designate the Annual schedule point.
6. Expand the **Weekly Schedules** area.
7. Enter the time values for each day, including Override 1 and Override 2.
8. When finished, click **Save Changes** at the top of the configuration tab.

Follow these rules when entering Weekly Schedule values.

- Enter time using the 24-hour clock format.
- When more than one *On* and *Off* time is entered for the same day, times must be entered in ascending order.
- Enter 00:01 (1 minute after midnight) to designate midnight.
- Do not leave empty days. If an *On* or *Off* state is to continue from a previous day, enter 00:01 in the first *On* or *Off* row for that day.
- Override1 and Override2 schedule times are used when the value controlled by the point entered in the Override Object 1 or Override Object 2 becomes *True* (1) or *On*.

Related topics

- *About KMD Schedules* on page 189
- *Configuring KMD Annual Schedule points* on page 191
- *Weekly Schedule points* on page 618

Configuring KMD Annual Schedule points

Use Annual Schedule points to designate special days of the year. Typically, these days are holidays, maintenance days, special occupancy days, or any other day that requires a schedule different than a weekly schedule. The state of an Annual Schedule Point is set to *True* when the date in the controller matches a scheduled date in the point. Usually, Annual Schedule points control the override schedule in a Weekly Schedule point.

To configure an Annual Schedule point, do the following:

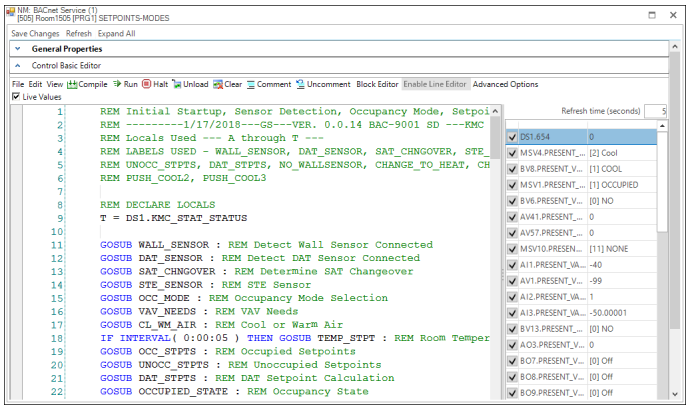
1. Open the Network Manager.
2. Expand the Network Manager list to reveal the controller, the Annual Schedules folder, and the Annual Schedule icons  in the folder.
3. Double-click the icon.
4. Under **General Properties**, enter the following:
   - Name
   - Description.
5. Expand the **Annual Schedule Dates** area.
6. Click a date on the calendar to designate it as a scheduled day. Click the date again to clear the selection.
7. Click **Save Changes** at the top of the configuration tab.

Related topics

- *About KMD Schedules* on page 189
- *Configuring KMD Weekly Schedule points* on page 190
- *Annual Schedule points* on page 589

KMC Connect

Part VI: Control Basic and the Code Editor



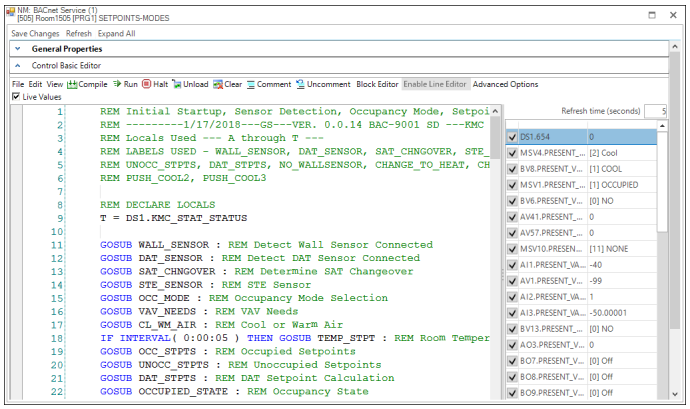
Section 27: Programming with the Code Editor

Topics in this section describe the Code Editor and using it to write Control Basic programs for KMC controllers.

The Code Editor is the programming tool with which Control Basic programs are entered and edited in the program objects in KMC controllers. Key features of the Code Editor include the following items.

- Editing programs within KMD or BACnet controllers from KMC Controls
- Writing and editing programs while offline
- Automatic Control Basic line numbering
- Color-coded text
- Automatic keyword display

Illustration 27–1 The KMC Connect Code Editor



Related topics

- *About Control Basic programs* on page 217
- *Writing Control Basic programs in controllers* on page 196
- *Writing programs offline* on page 198
- *Automatically numbering program lines* on page 200
- *Keyword prompting* on page 200
- *Using keyboard shortcuts* on page 201
- *Finding and replacing text* on page 202
- *Name/Mnemonic switching* on page 229
- *Changing Code Editor options* on page 206
- *Reading live values* on page 210

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.

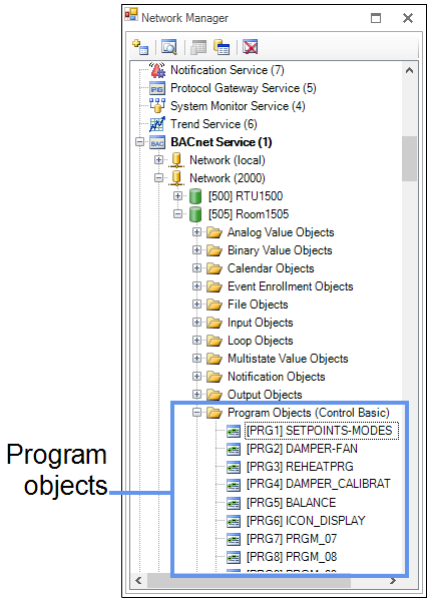
Note: Converting between block programming and line programming within the same program object is not recommended and usually is not possible. See the following topic, *Line vs. block programming* on page 218.

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 and then **Save**.

Tip: After compiling or saving a program, check the output window for errors in the program . The message or messages will include a line number and the cause of the error.

Illustration 27–2 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 a local program file by doing the following:

1 From the Code Editor toolbar, choose **File** and then **Save As**.

2 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 (continued)* on page 198.

Note: The Advanced Options are for special circumstances. Do not change unless directed by the KMC Controls Technical Support department.

Illustration 27–3 Code Editor toolbar

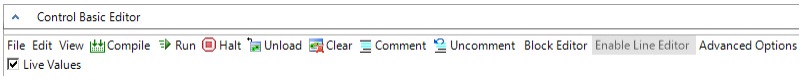


Table 27–1 Code Editor toolbar items

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 in the 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.
Name/Mnemonic switching		Switches object references between names and mnemonics in the Code Editor. See the topic *Name/Mnemonic switching* on page 229.
Clear (BACnet only)		Erases the Control Basic program from the controller.
Block Editor		Opens the Block Programming Editor

Table 27–1 Code Editor toolbar items (continued)

Action	Icon	Description
Enable Line Editor		Unlocks the Block Editor generated Control Basic for copying or editing with the line editor.
Live Values		See the topic *Reading live values* on page 210.

Related topics

- *About Control Basic programs* on page 217
- *Writing programs offline* on page 198
- *Automatically numbering program lines* on page 200
- *Keyword prompting* on page 200
- *Adding objects by dragging* on page 214
- *Using keyboard shortcuts* on page 201
- *Finding and replacing text* on page 202
- *Changing Code Editor options* on page 206
- *Writing block programs* on page 339
- *Name/Mnemonic switching* on page 229
- *Reading live values* on page 210

Writing programs offline

With the Code Editor, you can write Control Basic programs without connecting to a site or 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

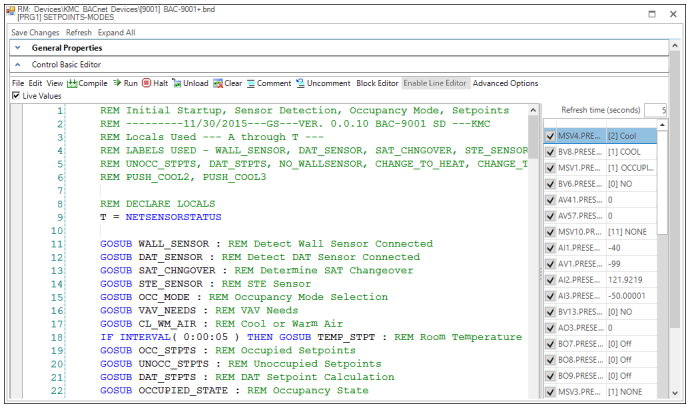
3 Add program statements and lines as required to operate the equipment connected to the controller.

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 27–4 A local file in the Code Editor



Edit an existing program

To use the Code Editor to modify local files — files with a `.cb`, `.bnd`, or `.kmd` extensions — 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 the 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 196
- *Automatically numbering program lines* on page 200
- *Keyword prompting* on page 200
- *Using keyboard shortcuts* on page 201
- *Finding and replacing text* on page 202
- *Changing Code Editor options* on page 206
- *Reading live values* on page 210

Automatically numbering program lines

The Code Editor will automatically add numbers to program lines.

To turn automatic line numbering on or off, do one of the following:

- From the Code Editor **View** menu, select or clear the **Auto Line Number** check box.
- 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 196
- *Writing programs offline* on page 198
- *Keyword prompting* on page 200
- *Using keyboard shortcuts* on page 201
- *Finding and replacing text* on page 202
- *Changing Code Editor options* on page 206
- *Reading live values* on page 210

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

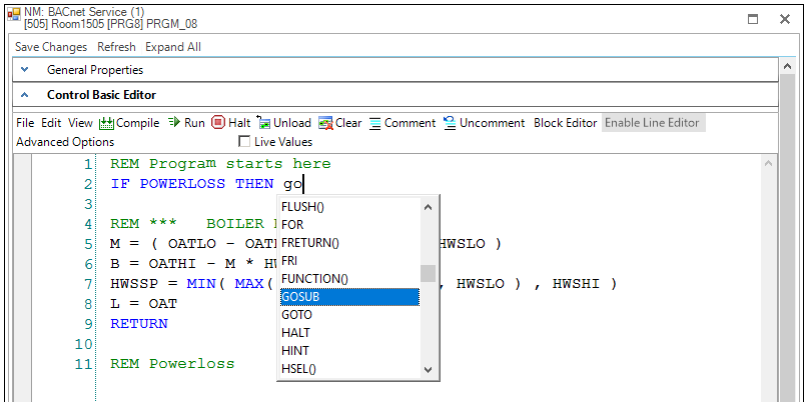
the list opens at the keyword *GOSUB*. You can then choose either *GOSUB* or *GOTO* from the list.

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 206.

Illustration 27–5 Keyword prompting list



Related topics

- *About Control Basic programs* on page 217
- *Writing Control Basic programs in controllers* on page 196
- *Writing programs offline* on page 198
- *Automatically numbering program lines* on page 200
- *Using keyboard shortcuts* on page 201
- *Finding and replacing text* on page 202
- *Changing Code Editor options* on page 206
- *Reading live values* on page 210

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 27–2 File menu shortcuts in Code Editor

Shortcut	Keyboard	Action
Renumber	Ctrl+**r**	Renumbers all lines in the program. Active only when automatic numbering is not enabled.
Undo	Ctrl+**z**	Reverses the last action.
Redo	Ctrl+**y**	Repeats the last 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.

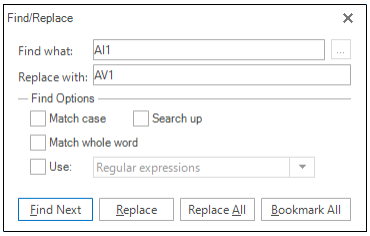
Related topics

- *About Control Basic programs* on page 217
- *Writing Control Basic programs in controllers* on page 196
- *Writing programs offline* on page 198
- *Automatically numbering program lines* on page 200
- *Keyword prompting* on page 200
- *Finding and replacing text* on page 202
- *Changing Code Editor options* on page 206
- *Reading live values* on page 210

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 27–6 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 for which to search.
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*.

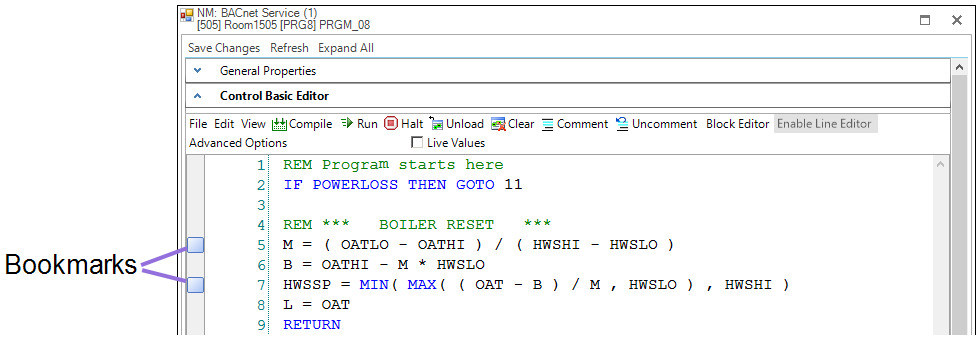
1. From the **Edit** menu, choose **Find And Replace**.
2. In the **Find what** box, enter the text for which to search.
3. In the **Replace with** box, enter the replacement text.
4. Select the check boxes for any other options.
5. Click **Find Next**, **Replace**, or **Replace All**.

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 for which to search.
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 27–7 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 characters are described in the table *Wildcard characters* on page 204.
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 27–3 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 205.

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 27–4 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 217

- *Writing Control Basic programs in controllers* on page 196
- *Writing programs offline* on page 198
- *Automatically numbering program lines* on page 200
- *Keyword prompting* on page 200
- *Using keyboard shortcuts* on page 201
- *Changing Code Editor options* on page 206
- *Reading live values* on page 210

Changing Code Editor options

The Code Editor options are located under three separate tabs.

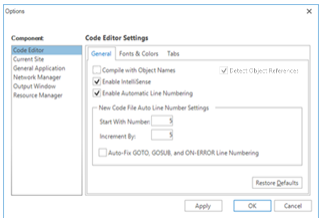
- The General Options tab controls Object Reference Detection, IntelliSense typing and line numbering options.
- The Fonts & Colors tab includes settings for the color and other text properties in the Code Editor window.
- Settings under Tabs tab control program formatting to enhance the readability of Control Basic.

To change the Code Editor appearance or options, do the following:

1. On the ribbon, click the **File** tab.
2. Click **Options**.
3. From the **Component** list, choose **Code Editor**.
4. Choose a tab to which to make changes in the **Code Editor Settings** dialog.
5. Click **Apply** or **OK** when finished.

General Options tab

Illustration 27–8 Code Editor Options dialog - General tab



Settings under the General Options tab control Object Reference Detection, IntelliSense typing and line numbering options.

Compile with Object Names When selected, object names may be used instead of mnemonics in Control Basic.

Detect Object References When selected, enables Name/Mnemonic object reference switching in the source code, color coding, and tooltip information. This is the default setting. The Detect Object References check box is selected automatically when the Compile with Object Names check box is selected. See *Name/Mnemonic switching* on page 229

Note: When the Compile with Object Names check box is cleared, the Detect Object References check box remains selected but is unavailable.

Enable IntelliSense When selected, keywords are verified as they are entered from the keyboard.

Enable Automatic Line Numbering When selected, automatically adds a line number at the start of a new line. The new line number is incremented by the value set in **Increment By**.

Start With Number When new pages open, the first line number is automatically set to the value in **Start With Number**.

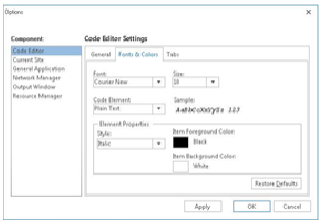
Increment By When **Turn Auto Line Number On** is selected, the Code Editor adds a line number every time a new line is started. The value of the line number is the previous line number plus the value of **Increment By**.

Auto Fix When selected, references to lines by a *GOTO*, *GOSUB*, or *ON-ERROR* are automatically updated when new line numbers are inserted.

Restore Defaults Returns all Code Editor options to the original settings.

Fonts & Colors tab

Illustration 27–9 Code Editor Options dialog - Fonts & Colors tab



Font, colors, and styles for text in the Code Editor window can be changed under the Fonts & Colors tab. Options are typical choices for text appearance in Windows programs.

Font Selects the typeface for the entire program.

Size Selects the point size for the entire program.

Code Element Selects the category of code element to modify.

Style Selects the style (e.g., regular, italic, bold) of the selected code element.

Item Foreground Color Selects the text color of the selected code element.

Item Background Color Selects the background color of the selected code element.

Restore Defaults Returns all text elements to their default settings.

Default text colors are listed in the table *Code Editor default text colors*.

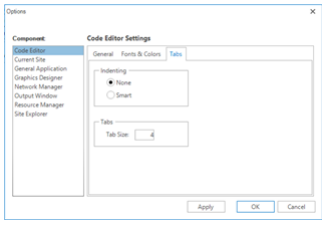
Table 27–5 Code Editor default text colors

Text function	Default color
Plain text	Black
Object references	Black
Comments	Green
Keywords	Blue
Strings	Maroon

Note: When entering text into the Code Editor, there is a 0.5 second delay before an object reference text changes color. This is part of the Control Basic scan that determines if the object reference is valid.

Tabs tab

Illustration 27–10 Code Editor Options dialog - Tabs tab



Settings under Tabs changes the way Code Editor adds tabs at the beginning of a new line. Tabs are invisible divisions that span five character spaces. 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

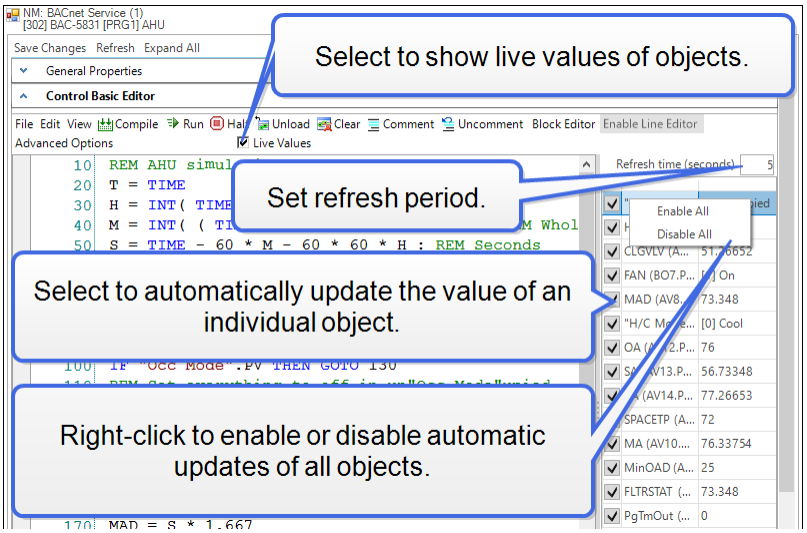
- *About Control Basic programs* on page 217
- *Writing Control Basic programs in controllers* on page 196
- *Writing programs offline* on page 198
- *Automatically numbering program lines* on page 200
- *Keyword prompting* on page 200
- *Using keyboard shortcuts* on page 201
- *Finding and replacing text* on page 202
- *Reading live values* on page 210

Reading live values

To view the actual present values of objects referenced from within a Control Basic program, use the Live Values option in the Code Editor.

- To display a list of objects used in the code and their most recent values, select the **Live Values** check box.
- To change the refresh rate, change the value in **Refresh time (seconds)** box.
- Clear the check box next to an individual object to prevent it from updating automatically.
- Right-click the object list to Enable or Disable automatic value updates for all objects.
- To change the order in the list, drag an object or group of objects to a new location.

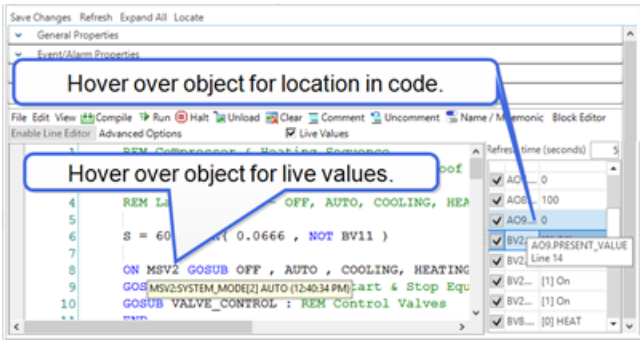
Illustration 27–11 Code editor live values



Hovering over an object displays additional information.

- Hovering over an object in the object list displays the line number(s) where the object is used in the program.
- When Live Values is enabled, hovering over an object in the code displays the object's mnemonic and name, its most recent value, and the time at which the value was read.
- When Live Values is disabled, hovering over an object in the code displays its mnemonic and name.

Illustration 27–12 Hovering for live values (when enabled) or location



Related topics

- *About Control Basic programs* on page 217
- *Writing Control Basic programs in controllers* on page 196
- *Writing programs offline* on page 198
- *Automatically numbering program lines* on page 200
- *Keyword prompting* on page 200
- *Using keyboard shortcuts* on page 201
- *Finding and replacing text* on page 202
- *Changing Code Editor options* on page 206

Locking Control Basic programs

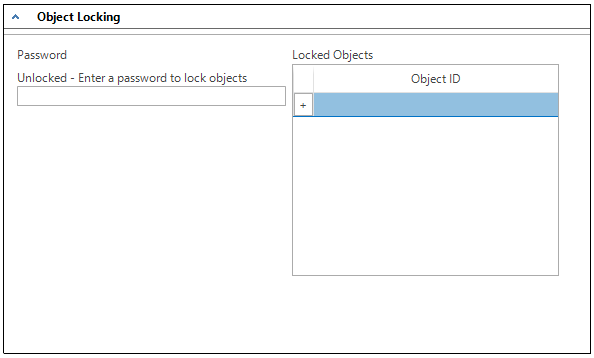
The Control Basic program in a BACnet Program object may be locked and hidden to prevent tampering. Locked files cannot be viewed, edited, or compiled with the Code Editor, but properties in the Program object are still accessible.

Caution: Locking a program requires a password. The password does not have a backdoor or recovery method. If it is unknown or forgotten, the program cannot be changed. The only recovery is to flash the controller firmware with the Firmware Upgrade Tool.

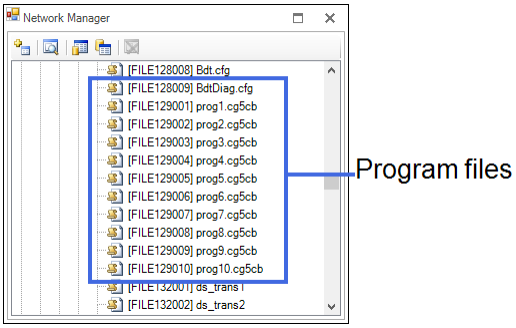
Locking a Control Basic program

To lock a Control Basic program, do the following:

1. In the Network Manager list, click ⊞ or ⊟ to expand or collapse the list to locate the BACnet device containing the program.
2. Double-click the device icon to open the Device Object configuration page in the workspace.
3. Expand the **Object Locking** area.



4. In the Network Manager list, click ⊞ to expand the device and locate the Files folder.
5. Click the ⊞ next to the Files folder.
6. In the Files folder, locate the Control Basic file. The file name format is similar to `[129005]prog5.cg5cb`. The number following "`prog`" in the extension is the same number as the Program object that contains the Control Basic program.



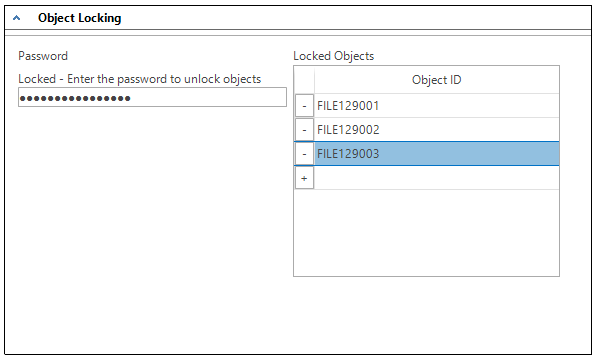
7. Drag the file icon to the Locked Objects list.
8. Enter a password. Do not lose this password; it cannot be recovered.
9. Click **Save Changes** at the top of the tab.

Unlocking a locked program

To unlock a Control Basic program, do the following:

1. In the Network Manager list, click ⊞ or ⊟ to expand or collapse the list to locate the BACnet device containing the program.
2. Double-click the device icon to open the Device Object configuration page in the workspace.

3 Expand the **Object Locking** area.



4 Enter the password.

5 Click the delete button - next to one or more file names.

6 Click **Save Changes** at the top of the tab.

7 Reenter the password.

8 Click **Save Changes** at the top of the tab.

See also the topic, *Device objects* on page 443.

Adding objects by dragging

Some types of BACnet objects may be dragged directly from the Network Manager or Resource Manager to the Code Editor for use in programs written for BACnet devices. This option can quickly add an object to or replace an existing object in a program.

Available object types are:

- Variables (Analog, Binary, Multistate)
- Inputs (Analog, Binary, Accumulator)
- Loops
- Outputs (Analog, Binary)

The following rules apply when dragging objects to the Code Editor.

- Objects may only be dragged to code written for the same device instance.
- Any text selected in the Code Editor must be cleared before dragging an object into a program.
- An object may be dragged to either:
  - A blank space in the code. This adds the object to the program.
  - An existing element in the code. The element is highlighted when a replacement is available. The user can choose whether this replaces a single instance or all instances of the object in the program.
- An object dragged into the Code Editor displays in Mnemonic format by default. To convert an object to Name format when dragging, press and hold the Alt key while releasing the mouse button at the desired position in the code.
- When dragging an object over text in the Code Editor, the pointer changes to a . An instance of the same object is highlighted as the pointer contacts it.
- Information about objects that have been replaced appears in the Output Window.

To add a new object to the code:

1. Open the program in the Code Editor.
2. Drag the object from the Network Manager or Resource Manager to the desired position in the program. If there is no text on the line, the object is inserted at the beginning of the line.

To replace a single instance of an object in the code:

1. Open the program in the Code Editor.
2. Drag the object from the Network Manager or Resource Manager to the desired position in the program.
3. In the dialog box, click **No**.

To replace all instances of an object in the code:

1. Open the program in the Code Editor.
2. Drag the object from the Network Manager or Resource Manager to the desired position in the program and press Ctrl.
3. In the dialog box, click **Yes**.

Related topics

- *Using the Network Manager* on page 41
- *About the Resource Manager* on page 57
- *Reference to BACnet objects* on page 435
- *About Control Basic programs* on page 217
- *Writing Control Basic programs in controllers* on page 196
- *Writing programs offline* on page 198
- *Automatically numbering program lines* on page 200
- *Keyword prompting* on page 200
- *Finding and replacing text* on page 202
- *Changing Code Editor options* on page 206
- *Reading live values* on page 210

Section 28: 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.

- *Line vs. block programming* on page 218
- *About Control Basic scans* on page 218
- *Programming format and notation* on page 222
- *Labels and line numbers* on page 220
- *Real numbers* on page 222
- *Hierarchy of operators* on page 223
- *Relational operators* on page 224
- *Using arithmetic operators* on page 225
- *Using Boolean logic* on page 225
- *Programming with variables* on page 226
- *Reading and writing BACnet objects* on page 227
- *Transferring values between BACnet controllers* on page 232
- *Reading and writing KMD points* on page 239
- *BACnet data types* on page 247
- *Generation 5 data types* on page 248
- *BACnet arrays and dynamic access* on page 246
- *User-defined functions and procedures* on page 244

Line vs. block programming

The recommend best practice for programming is to use either line or block programming and not attempt to switch between the two methods within the same program object. If a block program requires a function that is available only in line programming, do one of the following:

- In another program object, write the function using line programming and then share data between programs with value objects.
- Write the function inside of a Macro block. See the topic *Macro blocks* on page 347.

See the topic *Limitations and importing line programs* on page 350.

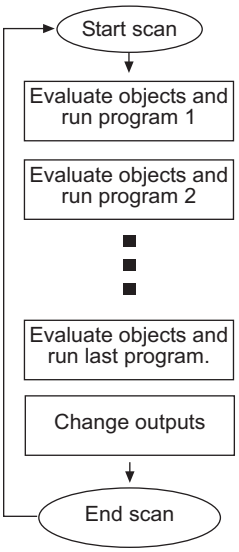
About Control Basic scans

Control Basic is the process that creates the automation in a KMC controller. Each controller has several program areas or 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 218 for an example of the scanning process.

Illustration 28–1 The scan process



The processor evaluates all program objects before making changes. For example, if programs 1, 3, and the last program includes instructions for Lights ON, and programs 2 and 4 have instructions for Lights OFF, the lights will not flash and they will be set to ON only 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

Control Basic programs are added to program objects and program points with the Control Basic editor.

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 types of statements.

Multiple statements

Multiple statements can be used on the same program line, but must be separated by a colon.

Example

```
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 as 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 functions and procedures* on page 244.

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 equate to a real number.

Table 28–1 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 697 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

Generation 5 Control Basic

- BAC-9000 series Conquest VAV controllers
- BAC-5900 series Conquest general purpose controllers
- BAC-9300 series Conquest unitary 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 and Generation 5 Control Basic do not use line numbers.

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 281
- *GOTO* on page 282
- *ONERROR* on page 305
- *ON GOSUB* on page 302
- *ON GOTO* on page 303

Declare a label by typing a name followed immediately by a colon (:).

- A label must start with a letter.
- 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 program lines. On each line there are one or more statements.

Throughout these instructions the following notations are used to describe formats:

Table 28–2 Typographic conventions

CAPS	Words in capital letters are keywords 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

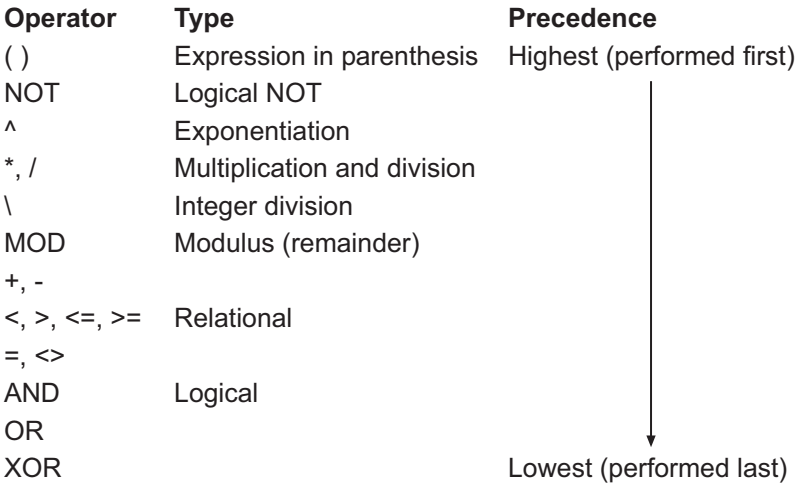
Real numbers are any logical number between $-3.4 * 10^{38}$ and $3.4 * 10^{38}$. Notation of a number is recognized in any of the following formats:

- 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 operations 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 223 when performing operations on an expression. Operations at the same level of precedence are evaluated from left to right.

Illustration 28–2 Order of operator precedence



Related topics

- *Using arithmetic operators* on page 225
- *Relational operators* on page 224
- *Using Boolean logic* on page 225

Relational operators

Relational operators are used to compare two values. The result is *TRUE* if the comparison is *TRUE*; otherwise, the result equals *FALSE*. The result can then be used to direct program flow. All relational operators have the same level of precedence.

Table 28–3 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 225
- *Using Boolean logic* on page 225
- *Hierarchy of operators* on page 223

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 225.

Table 28–4 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 224
- *Using Boolean logic* on page 225
- *Hierarchy of operators* on page 223

Using Boolean logic

Control Basic recognizes four Boolean 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 301.

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 257.

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 307.

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 338.

Related topics

- *Using arithmetic operators* on page 225
- *Relational operators* on page 224
- *Hierarchy of operators* on page 223

Programming with variables

Variables are placeholders for information such as setpoints, time delays, and operating modes. Control Basic uses two types of variables—local variables and value objects in BACnet controllers and variable points in KMD controllers.

Variables in KMD controllers

Variable points in KMD controllers are place holders for information such as setpoints, time delays, and modes. See the topic *Mnemonics for KMD controllers* on page 239 for using a variable point in Control Basic.

BACnet value objects as variables

Analog, binary, and multi-state 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 as 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. If a local variable is declared, single-letter variables must also be declared.

For details on using and declaring local variables, see the following keywords.

- *LOCALS* on page 292
- *INTEGER* on page 288
- *REAL* on page 314
- *STRING* on page 327

Reading and writing BACnet objects

Control Basic programs read from input objects and write to output objects by using mnemonics or—depending on the editor used to write the program—the name or description of the object. The programs can also read and write other objects such as schedule, value, and PID loop objects. In addition, Control Basic programs can read values from and write values to objects in other BACnet controllers on the same internetwork.

- For reference to the list of mnemonics, see *Mnemonics for BACnet controllers* on page 227.
- To read and write values in other controllers, see *Transferring values between BACnet controllers* on page 232.
- To read and write to the priority array and relinquish default, see *Reading and writing the priority array* on page 231.

Mnemonics for BACnet controllers

Mnemonics are short, easy-to-remember abbreviations in Control Basic that refer to BACnet objects. For example, a physical input is entered as AI1 or BI1 in BACnet controllers instead of typing *Input1*.



Note: The mnemonic codes are different between KMD and BACnet controllers. For KMD mnemonics see the topic *Mnemonics for KMD controllers* on page 239.

Object mnemonics Control Basic mnemonics for BACnet objects are listed in the table *BACnet object mnemonics* on page 228. 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* describe 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.
- Mnemonics are not case sensitive.



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 Program Object's Description of Halt property.

Table 28–5 BACnet object mnemonics

Object type	Mnemonic	Property	Action
Accumulator	ACC#	Present Value	Read Only
Analog Input	AI#	Present value	Read and Write
Analog Output	AO#	Present value	Read and Write
Analog Value	AV#	Present value	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
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#	Requires property mnemonic	

Property mnemonics The Next Generation and Generation 5 versions of Control Basic support reading and writing properties within objects. To designate a property, add the property mnemonic after the object number. The common properties are listed in the following table. The full name of the property—as seen in the user interface—may also work for some properties.



Note: Not all properties are supported in every object type.

Syntax: *object.property*

The following example enables trend log TL1 to start collecting data.

```
TL1.EN = 1
```

Table 28–6 Property mnemonics

Property	mnemonic	Action	Comments
High Limit	HL	Read and Write	Intrinsic alarming only
Loop Bias	B	Read and Write	Loop objects only
Loop Derivative	D	Read and Write	Loop objects only
Loop Integral	I	Read and Write	Loop objects only
Loop Proportional	P	Read and Write	Loop objects only
Loop Setpoint	SP	Read and Write	Loop objects only

Table 28–6 Property mnemonics (continued)

Property	mnemonic	Action	Comments
Low Limit	LL	Read and Write	Intrinsic alarming only
Present Value	PV	Read and Write	Default
Pulse Rate	PR	Read Only	Accumulator objects only
Relinquish Default	RD RELINQUISH_DEFAULT	Read and Write	Limit writes to changes only
Trend Log Enable	EN	Read and Write	Trend Log objects only

Name/Mnemonic switching

The Name/Mnemonic switching icon button () in the Control Basic Code Editor toolbar is used to toggle between name and mnemonic object references when writing, modifying and inspecting code. In order to use Name/Mnemonic switching, the Detect Object References option must be selected in Code Editor Settings. See *Changing Code Editor options* on page 206.

Name/Mnemonic switching is available in Generation 5, Next Generation and Standard versions of Control Basic.



Note: The Name/Mnemonic switching icon button is unavailable in the Code Editor when the Detect Object References option is not selected in Code Editor Settings or when using devices that are unsupported by Control Basic compilers.



Note: When Name/Mnemonic switching is active, moving the pointer over the same object in the Code Editor as the one selected in the Resource Manager or Network Manager highlights its background to aid in location.



Tip: Remember to click **Save** in order to update the program in the controller.

Generation 5 and Next Generation Control Basic implementation

Name/Mnemonic switching in Generation 5 Control Basic (KMC Conquest devices) and Next Generation Control Basic (FlexStat devices) can be applied to either a section of a program or to an entire program.

To convert a section of a program to the opposite object reference format:

1. Open the program in Code Editor.
2. Select the section of code including the complete object reference.

> **Note:** A complete object reference must be selected in order to use Name/Mnemonic switching in Generation 5 and Next Generation Control Basic. A partially selected object reference will not be converted.

3. In the Control Editor toolbar, click the **Name/Mnemonic** icon button .

To convert an entire program to the opposite object reference:

1. Open the program in Code Editor.
2. Click anywhere in the Code Editor workspace.
3. In the Control Editor toolbar, click the **Name/Mnemonic** icon button .

Standard Control Basic implementation

Name/Mnemonic switching in the Standard Control Basic used in Tier 2 BACnet (legacy KMC BACnet) and KMD (KMDigital) devices applies to the entire code. This means that one section of the program cannot be converted to the opposite object reference display format while the remainder of the program retains the original format.

To use Name/Mnemonic switching in Standard Control Basic:

1. Open the program in Code Editor.
2. Click anywhere in the workspace.
3. In the Control Editor toolbar, click the **Name/Mnemonic** icon button .

Related topics

- *Control Basic versions in controllers* on page 697
- *Mnemonics for BACnet controllers*
- *Mnemonics for KMD controllers*
- *About Control Basic programs*
- *Writing Control Basic statements* on page 219
- *Writing Control Basic programs in controllers* on page 196
- *Programming with the Code Editor* on page 195
- *Writing programs offline* on page 198

Reading and writing the priority array

The BACnet priority array is a method to maintain order when several commands are issued simultaneously to change the Present Value property of an object. When a command is issued, the object stores the value in its priority array at a designated priority level. The command with the highest priority controls the object. The array has 16 levels. Priority 1 is the highest and priority 16 is the lowest.

In addition, objects include a Relinquish Default property that is not part of the array. The value in Relinquish Default sets the Present Value if there are no values in the priority array.

Illustration 28–3 The BACnet Priority Array

	Priority Array	Present Value
1	Null	80%
2	Null	
3	Null	
4	Null	
5	Null	
6	Null	
7	Null	
8	80%	
9	Null	
10	60%	
11	Null	
12	Null	
13	Null	
14	Null	
15	Null	
15	Null	
16	Null	
Relinquish Default	50%	

Writing at a priority level The default for Control Basic is priority 9. To write to any other level, use an at sign (@) followed by the number of the priority level. The following example writes the value of 68 to Analog Output AO2 at priority level 8.

```
AO2@8 = 68
```

Relinquish a priority level To clear a value from a priority level, use the *RLQ* (relinquish) command.

```
RLQ AO1@8
RLQ BV7@7
```

Relinquish Default The value in the Relinquish Default property sets the value of the Present Value property if there are no values in the priority array. The value in Relinquish Default is persistent through a coldstart in the controller. Add the `RELINQUISH_DEFAULT` or `RD` mnemonic to an object mnemonic to read or write to the property.

The following example reads Relinquish Default in Analog Output object AO8.

```
AV14 = AO8.RD
```

To conserve the life of the FLASH memory in the controller, do not write to Relinquish Default on every scan. The following example writes to the Relinquish Default property only when the value at AI1 changes.

```
IF AI1 <> AV7 THEN AV7.RD = AI1
```

Transferring values between BACnet controllers

In a BACnet internetwork, controllers rarely operate without communicating with each other. In controllers from KMC Controls, Control Basic is used for this communication by reading from and writing to off-panel points. However, the rate of reading and writing off-panel points must be controlled to prevent flooding the network with unnecessary traffic. The method of controlling the rate of transfer will depend on the version of Control Basic that the controller supports.

The table *Versions of Control Basic* lists the version of Control Basic supported by each model of controller.

Topics in this section

Limits on transferring off-panel BACnet properties 232
Using WAIT statements to transfer values 233
Using NETPOINT and NETPOINTCACHE to transfer values 234
Reading and writing off-panel points with COV 235
Using WAIT statements to transfer values 237
Using INTERVAL for reading off-panel points 237

Limits on transferring off-panel BACnet properties

With Control Basic, a BACnet controller can request a property value from any other controller on the BACnet internetwork. The limit on the number of requests varies with each family of controllers.

BAC-5800 and BAC-7000 series controllers Each BAC-5800 or BAC-7000 series controller can request 128 values from no more than 32 devices.

Example: A single controller can make requests for off-panel values as shown in one of the following examples:

- Four values each from controllers 1-32
- Eight values each from controllers 1-16
- Sixteen values from controllers 1,5,6,8 and thirty-two values from controllers 9 and 10.



Note: In firmware earlier than R1.7.0.7 released September 2008, the number of requests is limited to 64 requests and 32 devices.

BAC-A1616BC, Flex Stat, and Conquest controllers Program no more than 256 requests for off-panel values in BAC-A1616BC, Flex Stat, and Conquest controllers.

See the following related topics for programming examples to read and write off-panel values.

- *Using WAIT statements to transfer values* on page 237
- *Using NETPOINT and NETPOINTCACHE to transfer values* on page 234
- The keyword *ALIAS* on page 254.

Using WAIT statements to transfer values

In controllers that support Standard Control Basic, use a WAIT statement to control the rate of reading and writing values between controllers Controlling the rate of reading and writing off-panel points prevents flooding the network with unnecessary traffic.

See also the keyword *WAIT* on page 336.

Reading properties from other BACnet devices To read a value from another panel, you must know the panel's device instance and the object mnemonic. The instance is separated from the value by a period (.). See the topic *Limits on transferring off-panel BACnet properties* on page 232 for the number of requests that can be programmed for each type of controller.

Syntax: *device instance.object*



Caution: In the following examples, the WAIT statements are required. Do not delete them or the program will not run correctly.

```
REM * Reading from Device 1213, AI7
AV24 = 1213.AI7 : REM Reading an off-panel value
WAIT 0:05:00
END
```

Writing values to other controllers To change the value in another BACnet device, you must know the instance number for the device. The device instance is separated from the object mnemonic by a period(.).

Syntax: *device instance.object[@priority]*

```
REM * Writing a value to Device 201, AV1
201.AV1 = AV1 : REM Writing an off-panel value
WAIT 0:05:00
END
```

Using NETPOINT and NETPOINTCACHE to transfer values

In controllers that support Generation 5 Control Basic, statements with NETPOINTCACHE and NETPOINT are an efficient method to limit the rate of reading from and writing to off-panel points. Controlling the rate of reading and writing off-panel points prevents flooding the network with unnecessary traffic.

Follow these basic rules when using NETPOINT and NETPOINTCACHE for off-panel points. Additional information can be found by looking in Help topics for the keywords.

- Always use NETPOINT and NETPOINTCACHE together.
- NETPOINT sets the interval for reading and writing off-panel points.
- NETPOINTCACHE enables a cache of values within a program from off-panel objects that are referenced with NETPOINT.

See also the keywords *NETPOINT* on page 297 and *NETPOINTCACHE* on page 298.

Syntax: *NETPOINT(device instance.object.property,read interval,write interval)*

- The intervals for reading and writing are expressed in seconds.
- If the point is read only then use a NAN in the write interval field.
- Within a controller, execute the NETPOINT command only once for each off-panel object.

Syntax: *NETPOINTCACHE(expression)*

- Enter one (1) for *expression* to enable NETPOINTCACHE and COV caching.
- Enter zero (0) for *expression* to disable NETPOINTCACHE and COV caching. This is the default state in controllers that support Generation 5 Control Basic.
- Add NETPOINTCACHE(1) to every program in the controller that uses an off-panel object that is referenced by the NETPOINT command.

The following examples include Control Basic programs that use NETPOINT and NETPOINTCACHE in a single program and a second example that uses NETPOINT in one program and NETPOINTCACHE in a second program.

Example 1: This program is an example of reading one off-panel point.

```
NETPOINTCACHE(1):REM REM This Turns on the point caching
for points in this program.
IF POWERLOSS=0 THEN
    NETPOINT(5.AV1.PV,300,NAN) : REM Cache is updated
    every 300 seconds (5 MINUTES).
ENDIF
IF 5.AV1 > 50 THEN START BV1 : REM The value of 5.AV1
will be 0 to 300sec old
END
```

Example 2: This is an example of retrieving off-panel points in Program 1 and referencing them in Program 2.

Program 1

```
IF POWERLOSS=0 THEN
    NETPOINT(5.AV1.PV,300,NAN)
    NETPOINT(6.AV1.PV,200,NAN)
ENDIF
END
```

Program 2

```
NETPOINTCACHE(1)
IF 5.AV1 > 50 THEN START BV1 : REM The value of 5.AV1
will be 0 to 300sec old
END
```

Reading and writing off-panel points with COV

In controllers that support Generation 5 Control Basic, a COV (Change Of Value) subscription transfers data between controllers only when there is a predetermined change to a remote value. It is a quick method to receive an update and still limit unnecessary network traffic. Controlling the rate of reading and writing off-panel objects prevents flooding the network with unnecessary traffic.

Follow these basic rules when using COV for off-panel points. Additional information can be found by looking in Help for the keyword *COV* on page 266.

- Within a controller, execute the COV command only once for each off-panel object.
- Add NETPOINTCACHE to every program in the controller that uses the off-panel object that is referenced by the COV command.
- For redundancy, combine COV with NETPOINT to periodically update the off-panel value regardless of change.
- Use ERRORLEVEL to detect when a COV subscription is not returning a valid value from the off-panel point.

See also the keywords *NETPOINT* on page 297, *NETPOINTCACHE* on page 298, and *ERRORLEVEL*.

Syntax: *COV(device.object.property, lifetime, confirmed, increment)*

Two Control Basic examples follow: Example 1 is a single program that sets up the COV subscription and references the off-panel object in a single program. Example 2 sets up the subscription in one program and references the object in a second program.

Example 1: In this example, a COV subscription is set up in the same program that uses the the remote value. When Binary Value object 1 in Device Instance 1 (the remote device) changes from *TRUE(1)* to *FALSE(0)* or *FALSE(0)* to *TRUE(1)*, the value is transferred to the local controller. Then, the local controller uses that value to start or stop the BV1. To provide redundancy, NETPOINT automatically reads BV1 every 300 seconds (5 minutes). The statements with COV and NETPOINT run only during the first scan after a restart.

When using the Control Basic COV method, add error detection with ERRORLEVEL. When the program runs the first time and there is reference to off-panel object BV1, the value is returned as *NAN* (Not A Number). The value will continue to be returned as NAN until a valid subscription is established. In addition, NAN is returned if the program requests the off-panel value and, after three attempts, does not receive a reply. When a valid number is returned, ERRORLEVEL is *FALSE*(0) and the COV value for the off-panel BV1 is used.

```
NETPOINTCACHE(1)
IF POWERLOSS THEN
    NETPOINT(5.BV1.PV,300,NAN)
    COV(5.BV1.PV,300,0,1)
ENDIF

IF 5.BV1=1 AND ERRORLEVEL=0 THEN START BV1 ELSE BV1=0
END
```

Example 2: This is an example of initializing a COV subscription to an off-panel point in Program 1 and referencing the point in Program 2. The logic is similar to Example 1 except that Analog Value Object 2 in Device Instance 605 is subscribed for a Change of Value of 3.0.

Program 1

```
NETPOINTCACHE(1)
IF POWERLOSS THEN
    NETPOINT(605.AV2.PV,300,NAN)
    COV(605.AV2.PV,300,0,3.0)
ENDIF
```

Program 2

```
NETPOINTCACHE(1)
IF ERRORLEVEL=0 THEN av7=605.AV2
```

Using WAIT statements to transfer values

In controllers that support Standard Control Basic, use a WAIT statement to control the rate of reading and writing values between controllers Controlling the rate of reading and writing off-panel points prevents flooding the network with unnecessary traffic.

See also the keyword *WAIT* on page 336.

Reading properties from other BACnet devices To read a value from another panel, you must know the panel's device instance and the object mnemonic. The instance is separated from the value by a period (.). See the topic *Limits on transferring off-panel BACnet properties* on page 232 for the number of requests that can be programmed for each type of controller.

Syntax: *device instance.object*



Caution: In the following examples, the WAIT statements are required. Do not delete them or the program will not run correctly.

```
REM * Reading from Device 1213, AI7
AV24 = 1213.AI7 : REM Reading an off-panel value
WAIT 0:05:00
END
```

Writing values to other controllers To change the value in another BACnet device, you must know the instance number for the device. The device instance is separated from the object mnemonic by a period(.).

Syntax: *device instance.object[@priority]*

```
REM * Writing a value to Device 201, AV1
201.AV1 = AV1 : REM Writing an off-panel value
WAIT 0:05:00
END
```

Using INTERVAL for reading off-panel points

In Standard Control Basic, use INTERVAL with an IF THEN statement to control the rate of reading off-panel points. Controlling the rate of reading and writing off-panel objects prevents flooding the network with unnecessary traffic.

Example 1: Without error checking

This example is the simplest form of using INTERVAL and assumes that the controller will always respond correctly. It reads, at one minute intervals, the value of Analog Input AI4 in controller Device Instance 4410. The value of AI4 is then assigned to Analog Value object AV24 in the controller running the program.

```
IF INTERVAL ( 00:01:00 ) THEN AV24 = 4410.AI4
```

Example 2: Error checking

The program in Example 1 assumes that the controller will always respond correctly to a request for a value. However, if a controller is off-line or the controller's Device Instance changes, an error will result when using INTERVAL.

The program in this example begins by reading, at one minute intervals, the value of Analog Input object AI4 in controller Device Instance 4410.

If the program does not detect an error reading the off-panel point, it uses the value from 4410.AI4 and jumps to label NO_ERROR to continue evaluation.

If an error is detected, the program jumps to the label FOUND_ERROR and assigns the fixed value "70" to AV24. It then continues to run the program as if no error took place.

Next Generation and Generation 5 Control Basic

```
IF INTERVAL ( 00:01:00 ) THEN AV24 = 4410.AI4
ONERROR FOUND_ERROR
GOTO NO_ERROR : REM Jump around error recovery
FOUND_ERROR:
REM Error recovery
AV24 = 70
NO_ERROR: REM Continue program
```

Standard Control Basic

```
10 IF INTERVAL( 0:01:00 ) THEN AV_24 = 4410.AI4
20 ONERROR 40
30 GOTO 50 : REM Jump around error recovery
40 AV_24 = 70 : REM Error recovery
50 REM Continue program
```

Example 3: Validating the value

This example is similar to Example 2 except that it tests to see if the received value is a legitimate BACnet number. The keyword ISNAN tests 4410.AV4 and if it is not valid, substitutes the value of "55". If 4410.AV4 is valid, the program continues at ELSE and the off-panel value is used. This example is not supported for Standard Control Basic.

```
IF INTERVAL ( 00:01:00 ) THEN
REM Verify that the value is good
    IF ISNAN( 4410.AI4 ) THEN
        REM Set a default value
        AV24 = 55
    ELSE
        REM Use the received value
        AV24 = 4410.AI4
    ENDIF
ENDIF
```

Reading and writing KMD points

Control Basic programs read from and write to points by using mnemonics or—depending on the editor used to write the program—the name or description of the point. The types of points include input and output points as well as points such as variables, schedules, or PID loops. In addition, Control Basic programs can read the value of points in other KMD controllers on the same network.

- For reference to the list of mnemonics for KMD controllers, see *Mnemonics for KMD controllers* on page 239.
- To read values from other controllers, see *Transferring values with KMD want-points* on page 240.

Mnemonics for KMD controllers

Mnemonics are short, easy-to-remember abbreviations in Control Basic that refer to points. For example, a physical input is entered as *IN1* in KMD controllers.



Note: The mnemonic codes are different between KMD and BACnet controllers.

Control Basic mnemonics for KMD points are listed in the table *Control Basic Mnemonics for KMD points* on page 240. The following line of Control Basic is an example of using mnemonics to refer to an analog input point and a digital output point.

```
10 If IN1 < 70 then start out2
```

- Mnemonics are reserved words and cannot be used for point labels or descriptions.
- The column *Read Field* describes attributes that can be read from that point type, such as its value or status.
- The column *Write Field* describes attributes that may be changed on a point type through programming or by direct access, such as its value or status.
- The total time in a Runtime Log point is automatically rounded to whole hours.

Table 28–7 Control Basic Mnemonics for KMD points

Point Type	Mnemonic	Read Field	Write Field
Annual Schedules	AS#	Status	N/A
Array Elements*	AY#	Value	Value
System Groups	GRP#	N/A	N/A
Inputs	IN#	Value	Value
Outputs	OUT#	Value	Value
PID Controllers	CON#	Output Value	Setpoint Value
Programs	PRG#		Status
Runtime Logs	RT#	ON TIME	N/A
Trend Logs	TL#	Status	Status
Variables	VAR#	Value	Value
Weekly Schedules	WS#	Status	N/A

* Array Elements are only available in Tier 1 controllers.

Related topics

- *Transferring values with KMD want-points* on page 240
- *KMD want-point limitations* on page 241

Transferring values with KMD want-points

A Control Basic statement can request the value of points from other KMD controllers on the same network. Request for these points—referred to as want-points—follow these general rules.

- A Control Basic statement can only request data from other controllers; it cannot command a want-point.
- Tier 1 controllers can request points from other Tier 1 controllers connected to the same KMD network and Tier 2 controllers connected to it.
- Tier 2 controllers can only request points from other controllers on the same Tier 2 network and from the Tier 1 controller to which the network is connected.
- The initial want-point value is set to zero.
- A controller will hold the last known value of a want-point even if the controller that is sending the point is off-line.

See also the topic, *KMD want-point limitations* on page 241 for restrictions on the number of want-points that can be requested.

Syntax for want-points

To designate a want-point, add the controller panel number to the front of the mnemonic. Use the same Control Basic mnemonics that designate local points to designate want-points. The exact format will depend on whether the want-point request comes from within a Tier 1 or Tier 2 controller. See *Mnemonics for KMD controllers* on page 239.

Control Basic want-point examples

The format for Tier-1 and Tier 2 want-points is slightly different. Examples are shown for each type of controller.

Tier 1 examples In Tier 1 controllers, refer to Tier 2 controllers by their network letter and panel number. Model KMD-5210, the Lan Controller, has both A and B networks. KMD-5205 and KMD-5270 have only an A network but, the letter A must be used.

```
10 VAR3 = A4OUT3
20 VAR4 = B4IN4
```

Refer to points in other Tier 1 controllers by their panel number.

```
10 VAR5 = 4IN6
```

Tier 2 examples In the Control Basic running in a Tier 2 controller, use address 0 (zero) for the Tier 1 controller regardless of the controller's panel number.

```
10 VAR1 = 0IN1
```

For Tier 2 controllers on the same Tier 2 network, use the remote controller's panel number.

```
10 VAR2 = 1OUT2
```

KMD want-point limitations

KMC direct digital controllers read values from points in controllers across a network by requesting want-points. These requests create a want-point list that is filled and refreshed as each controller puts data on the network. However, there are restrictions on the number of points that can be transferred.

- *Tier-to-Tier transfer limitations*
- *Total want-points transferred*
- *Tier 2 extended points*
- *Transfer time*

Tier-to-Tier transfer limitations

The following table lists the number of want-points that may be transferred between any two controllers.



Note: Extended points may not be transferred from tier-to-tier. See *Tier 2 extended points* on page 243 for additional information about points transferred between Tier 1 and Tier 2 controllers.

Table 28–8 Maximum single controller tier to tier want-points

Point Type	Tier 2 to Tier 1	Tier 1 to Tier 2
Outputs	8	64
Inputs	8	64
Variables	32	64
Weekly Schedules	4	32
Annual Schedules	2	8
Runtime Logs Total Hours	8	64
PID Control loops	8	64
System Groups	4	64

Total want-points transferred

The total number of want-points from all sources that can be transferred into or out of a single Tier 1 controller.

Table 28–9 Total Tier 1 controller want-points

Tier	In	Out
Tier 1	127	127
Tier 2	512	64

The following table lists the total number of want-points from all sources that can be transferred into or out of a single Tier 2 controller.

Table 28–10 Total Tier 2 controller want-points

Controller	In	Out
KMD-55xx	32	32
KMD-6000	32	32
KMD-5801/02	124	32
KMD-5821	124	32
KMD-5831	124	63
KMD-7xxx	32	32

Tier 2 extended points

The KMD-5800 series of controllers have points not found in the original generation of KMD controllers. These additional points are referred to as *extended points* and can only be transferred as follows:

- Extended points *can* be shared across the network with other KMD-5800 or KMD-7000 series controllers.
- Extended points *cannot* be shared with KMD-5500 or KMD-6000 series controllers.
- Extended points *cannot* be shared with attached Tier 1 controllers.

The table *KMD-5800 Series extended points* on page 243 summarizes the extended points found in the KMD–5800 series controllers.

> **Tip:** To transfer an extended point to a Tier 1 controller (either a KMD-5100 or KMD-5200 series controller), assign the value to a variable within the source controller and then share the variable with the Tier 1 controller.

Table 28–11 KMD–5800 Series extended points

Controller	Input	Output	Variables	PID Control Loops	System Groups	Weekly Schedules	Annual Schedules
KMD-5801			33-64				
KMD-5802			33-64				
KMD-5821			33-64				
KMD-5831	9-16	9-12	33-128	9-16	5-8	5-8	3-4

Transfer time

The time to transfer want-points depends on the type of controller and firmware version.

Tier 1 Tier 1 controllers send want-points at regular intervals. The interval depends upon the version of firmware in the controller.

Firmware build 1.213 and later want-points are transferred every 20-25 seconds.

Prior to firmware build 1.213 want-point transfer time is calculated as follows:

want-points = Number of controllers x 3

Tier 2 A Tier 2 want-point list is not immediately filled or refreshed. Each time the token is passed to a controller, the protocol permits the controller to transmit up to eight points. A full controller, with 32 points to put onto the network, will have all of its information passed only every fourth time the token passes by.

User-defined functions and procedures

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 programming steps within a Control Basic program.



Note: This applies only to controllers with Generation 5 Control Basic.

See the related keywords *PROCEDURE* on page 312 and *FUNCTION* on page 280.

Rules for functions and procedures.

- All functions and procedures must have unique names. The names cannot duplicate Control Basic keywords, 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 performs 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 that 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
```

BACnet 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 697.

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 in 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 292
- *INTEGER* on page 288
- *REAL* on page 314
- *STRING* on page 327

BACnet 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 performed 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 288
- *STRING* on page 327
- *REAL* on page 314
- *Generation 5 data types* on page 248

Generation 5 data types

To maintain compatibility with current and future BACnet standards, the Generation 5 version of Control Basic expands the use of data types. This includes support for String, Signed Integer, and Unsigned Integer data types. This expansion also changes the way Control Basic uses data type Real. This results in some issues with division and lone-value variable assignment.

Division issues When performing division, if either the dividend or the divisor is data type Real, the result—the quotient—is also data type Real. However, if both parts of the division calculation are data type Integer, the result is data type Integer. This results in an error when the division is used with a BACnet property that is data type Real, for example the Present Value property in an Analog Value object. The following statement produces unexpected results because the Present Value in AV1 is expecting data type Real, but the division operation quotient is data type Integer.

```
AV1=1/2
```

To correct the calculation, add a decimal point to one or both of the values.

```
AV1=1.0/2     AV1=1/2.0     AV1=1.0/2.0
```

Lone-value variables The data type issue is found also with the assignment of lone-value variables. Lone-value variables are the variables in an equation whose only purpose is assigning a value to the variable. The following example produces the same error previously shown for division.

```
A=1
B=2
AV1=A/B
```

The correction is the same as with the division error. Add a decimal point to one of the variables.

```
A=1.0         A=1           A=1.0
B=2           B=2.0         B=2.0
AV1=A/B       AV1=A/B       AV1=A/B
```

Complex equations For complex equations—equations that include several operations—some parts of the equations may be calculated as data type Integer and other parts as data type Real. When the equation is evaluated, the data type for the result is determined by the order that the equation is evaluated. In the following example, 1/2 is evaluated first and sets the data type as Integer. Then, the result is divided by 4.0, but because the data type is already set as data type Integer, the result is also data type Integer even through there is a decimal point in the number.

```
AV1=1/2/4.0
```

Correct the problem by adding a decimal to the part of the equation that is evaluated first by Control Basic.

```
AV1=1/2.0/4
```

Summary of the rules

- Division results in a data type of Integer unless a decimal point is added to a number in the equation.
- Lone-value variable assignment (a=1) is data type Integer.
- In complex equations, the data type established first is the data type for the entire result. The hierarchy of evaluation is parentheses, exponents, division, multiplication, addition, subtraction.
- If either part of a calculation is data type String, then the other side is converted to data type String.
- If either part of a calculation is data type Real, then the other side is converted to data type Real.
- If either part of a calculation is data type Integer, then the other side is converted to data type Integer.

See also the topic *BACnet data types* on page 247 and *Hierarchy of operators* on page 223.

Section 29: Keywords for Control Basic

This section covers the keywords for the Control Basic programming language.

Control Basic reserves keywords for operators, commands, and functions. These keywords may not be used for descriptors, labels or names of points, objects, variables, or procedures.

To refer to KMC points in Control Basic programs, see the topic *Mnemonics for KMD controllers* on page 239.

To refer to BACnet objects in Control Basic programs, see the topic *Mnemonics for BACnet controllers* on page 227.

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)
```

KMD example

Returns *2.3*, the absolute value of -2.3.

```
A = ABS( -2.3 )
```

Returns the absolute value of the result of the calculation.

```
C = ABS( SETPOINT - SPACETMP )
```

ALARM

The ALARM statement initiates an alarm notification from a KMD controller. It may also initiate a dialing sequence in a modem connected directly to the controller in which the alarm notification is created.

Syntax: *ALARM_expression_,_differential_,_string*



Note: For KMDigital controllers only.

Expression The *expression* parameter defines a condition that triggers the alarm. It is expressed as a conditional such as X_<_Y or X_>_Y or X_=_Y. The values for X and Y can be any number, variable, input, or output in any controller. Spaces within the expression are required and are indicated by the underscore (_).

Differential The *differential* parameter defines a dead band. The alarm returns to normal after the condition is outside of the dead band. Using *differential* minimizes multiple alarms when conditions change rapidly between alarm conditions.

String This is the text message that is added to the alarm viewer list. The message may be up to 69 characters long.

The following examples sets up alarms when the temperature measured at Input *IN1* drops below 34 degrees. Once triggered, the alarm will not clear until the air temperature rises to or above 37 (34+3) degrees. The differential value eliminates multiple alarms if the temperature cycles rapidly between 33.9 and 34.0 degrees.

```
ALARM IN1 < 34 , 3 , Air temperature is close to
freezing.
```

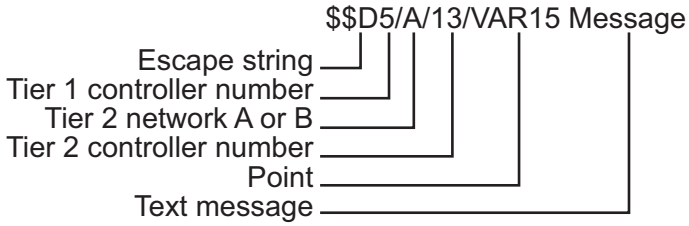
Use the following alarm message for TotalControl alarms. The text immediately following *$$D* indicates the address of the controller that generated the alarm. This is explained in detail under *Setting up routing for TotalControl alarms* on page 252.

```
ALARM IN1 < 34 , 3 , $$D5/A/13/IN1 Air temperature is
close to freezing.
```

Setting up routing for TotalControl alarms

Without TotalControl alarm routing, the text defined by *string* will be added to the alarm viewer but not the point of the alarms origin. To use TotalControl alarm routing, you must add to *string* information about the point that originated the alarm. The format is shown in the illustration, *ALARM* on page 252.

Illustration 29–1 Alarm routing message string for KMD controllers




```
ALARM VAR15 < 68 , 0 , $$D1/A/13/VAR15 Gym below
setpoint!
```

For WinControl XL email routing

To duplicate the WinControl XL Plus email feature, create one or more alarm classes named *E1-E8*.

Add the alarm class to the alarm message string as shown in the following example:

```
ALARM IN4 > 85 , 3 , E1 Area temp too high
```

For pager notification

By adding the CALL, NPAGE, or TPAGE options to the ALARM text string, the controller will initiate a modem dialing sequence for pagers or another computer running WinControl XL. The KMD Tier 1 controller that initiates the alarm must have a modem connected to it for pager notification.

Table 29–1 Pager message string options

Alarm device	String option
Numeric pager	NPAGE(_1234567_)_5555555
Text pager	TPAGE(_1234567-1234_)_Message

```
ALARM AIR/TP < 34 , 3 , NPAGE( 1234567 ) 5555555
ALARM AIR/TP < 34 , 3 , TPAGE( 1234567 1234 ) Message
```

Related topics

- *DALARM* on page 268
- *NPAGE* on page 301
- *PHONE* on page 309
- *TPAGE* on page 335

ALIAS

Use ALIAS to read from and write to off-panel objects. ALIAS declares a local variable and dynamically binds the value of a property to that variable. The syntax for ALIAS is slightly different between Next Generation and Generation 5 Control Basic.

- For the KMC Controls Building Controller and FlexStat, see *ALIAS for Next Generation Control Basic*.
- For Conquest controllers, see *ALIAS for Generation 5 Control Basic* on page 256.

See also the related topic *FLUSH* on page 279.

ALIAS for Next Generation Control Basic

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[priority], local, read interval, write interval)*

Note: Applies only to controllers with Next Generation Control Basic.

The same point may be bound to an ALIAS statement in more than one program. However, the shortest read or write interval of all ALIAS statements within the device is used in all programs.

Table 29–2 Parameter for Next Generation ALIAS

Parameter	Description	Comments
device	The device instance number or name.	Enclose the name of a device with quotation marks (" "). Names are case sensitive.
object	A valid mnemonic	See *Mnemonics for BACnet controllers* on page 227.
property	The property of the off-panel object.	
priority	The priority for writing.	If no priority is designated, the default value is @9. Priority is ignored for objects without a priority array such as inputs.
local	The local variable to use within the program	Use as a local variable within the Control Basic program in which ALIAS is declared.

Table 29–2 Parameter for Next Generation ALIAS (continued)

Parameter	Description	Comments
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*.

Reading off-panel objects The following Control Basic example binds the present value of Analog Input object 1 in device 1212 to the local name *LocalOAT*. Control Basic reads the value of off-panel object AI1 every 10 minutes. The value of *LocalOAT* is then used to set the value of local object AV1.

```
ALIAS(1212, AI1, PV, LocalOAT, 00:10:00, NONE)
AV1=LocalOAT
```

Writing to off-panel objects The following Control Basic example binds the present value of Binary Value object BV1 in the device named "RTU" to the local name *GlobalOccupied*. Control Basic writes the state of the local Binary Value object BV1 to the off-panel BV1 every four minutes.

```
ALIAS("RTU", BV1, PV@8, GlobalOccupied, NONE, 00:04:00)
GlobalOccupied=BV1
```

The last example binds the value of the Analog Input object AI2 to the local variable *OutsideAirTemp*. When the device name 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)
AV1=OutsideAirTemp
```

ALIAS for Generation 5 Control Basic

This statement declares a local variable and dynamically binds the value of a property to that variable.

Syntax: *ALIAS(device, object, property[priority], local)*

Note: Generation 5 Control Basic only.

Note: For accessing off-panel points with Generation 5 Control Basic compatible controllers, use either *NETPOINT* or *COV* on page 266.

The same point may be bound to an ALIAS in more than one program. See also the related topic *FLUSH* on page 279.

Table 29–3 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 *Mnemonics for BACnet controllers* on page 227.
property	The property of the off-panel object.	
priority	The priority for writing.	If no priority is designated, the default value is @9. Priority is ignored for objects without a priority array 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 225.

ARCCOS

Returns the arccosine of the specified angle. *Angle* is expressed in radians.

Syntax: *ARCCOS(angle)*



Note: For BACnet controllers only. See *COS-1* on page 266 for KMD controllers.

```
A = ARCCOS( AV1 )
```

See the related topic *COS* on page 266.

ARCSIN

Returns the arcsine of the specified angle. *Angle* is expressed in radians.

Syntax: *ARCSIN(angle)*



Note: For BACnet controllers only. See *SIN-1* on page 323 for KMD controllers.

```
A = ARCSIN( AI8 )
```

See the related topic *SIN* on page 324.

ARCTAN

Returns the arctangent of the specified angle. *Angle* is expressed in radians.

Syntax: *ARCTAN(angle)*



Note: For BACnet controllers only. See *TAN-1* on page 328 for KMD controllers.

```
A = ARCTAN( AV12 )
```

See the related topic *TAN* on page 328.

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)
```

KMD example

```
D = AVG( IN1 , IN3 , IN6 )
```

BAC-GET

Returns the present value from an object in a BACnet device. The BACnet *device instance* number must be in the range from 1 to 4,194,303. The BACnet *object* is composed of the object type and the object number.

Syntax: *BAC-GET (_device number_,_object_)*



Note: BACnet licensed KMD Tier 1 controllers only.

Table 29–4 Supported BACnet object types

Object type	Mnemonic to use
Analog Input	AI
Analog Output	AO
Binary Input	BI
Binary Output	BO
Analog Value	AV
Binary Value	BV

In the following example, ON-ERROR redirects the program if a BACnet error occurs. *WAIT* on line 40 halts program execution while the BACnet device responds to the program.

```
A = BAC-GET( 1 , AV01 )
ON-ERROR 40
VAR10 = A
WAIT 0:00:30 : REM Line 40
```

BAC-RLQ

BAC-RLQ relinquishes control to the specified priority to the object in the BACnet device. The BACnet *device instance* number must be in the range from 1 to 4,194,303. The BACnet *object* is composed of the object type (see the table *Supported BACnet object types* on page 259) and the object number. The priority number must be in the range of P1 to P16. See the table *Standard BACnet priorities* on page 260 for the BACnet standard priority levels.

Syntax: *BAC-RLQ(device instance , object , priority)*



Note: For BACnet licensed KMD Tier 1 controllers only.

Table 29–5 Standard BACnet 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/Off time
P7	
P8	Manual Operator
P9	Default for Control Basic
P10–P16	

In the following example, ON-ERROR redirects the program if a BACnet error occurs. The WAIT statement halts program execution while waiting on a response from the BACnet device.

```
BAC-RLQ( 54321 , AO14 , P12 )
ON-ERROR 40
WAIT 0:00:30
REM Program continues at Line 40
```

BAC-SET

This command sends a value at the specified priority, to a point in a BACnet device. The BACnet *device instance* number must be in the range from 1 to 4,194,303. The BACnet *object* is composed of the object type (see the table *Supported BACnet object types on page 259*) and the object number.

Syntax: *BAC-SET (device instance , object , priority , value)*

Note: BACnet licensed KMD Tier 1 controllers only.

In the following example, ON-ERROR redirects the program if a BACnet error occurs. WAIT on line 30 halts program execution while the BACnet device responds to the program.

```
BAC-SET( 5 , BO14 , P9 , VAR10 )
ON-ERROR 40
WAIT 0:00:30
REM Continue program here at Line 40
```

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 29–6 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 29–7 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 )
```

BUILD-NUMBER

This function returns the firmware version number stored in the controller.



Note: For KMD controllers only.

```
VAR10 = BUILD-NUMBER
```

CLEAR

Resets the value of all local variables—variables labeled A-Z and declared variables—to zero.

```
CLEAR
```

CLEAR-COUNT

Resets the error count in a KMD Tier 2 controller to zero.



Note: For KMD Tier 2 controllers only.

```
A = CLEAR-COUNT
```

See the related topic *ERROR-COUNT* on page 278.

CLOSE

Sets the value of a named point, KMD variable, 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
```

KMD example

```
CLOSE VAR1
CLOSE A
```

Related topics

- *OPEN* on page 306
- *START* on page 325
- *STOP* on page 326

CONBIAS

CONBIAS sets the bias property of a PID control loop. *PIDcontroller* is the number of the PID control loop within the controller. *Expression* is the new bias value within the range of 0–100. The value for *Expression* can also be a variable or a computation.



Note: For KMD controllers only. For BACnet controllers, see the topic *Mnemonics for BACnet controllers* on page 227.

Syntax: *CONBIAS(_PIDcontroller_,_Expression_)*

```
CONBIAS( 1 , 1.1 )
```

Related topics

- *CONPROP* on page 264
- *CONRATE* on page 264
- *CONRESET* on page 265

CONPROP

CONPROP sets the proportional band of a PID control loop. *PIDcontroller* is the PID control loop number. *Expression* is the new proportional band value within the range of 0-4000. The expression can also be a variable or a computation.

Note: For KMD controllers only. For BACnet controllers, see the topic *Mnemonics for BACnet controllers* on page 227.

Syntax: *CONPROP(_PIDcontroller_,_Expression_)*

In the following example, if VAR1 is less than 3, the proportional band on controller #1 will be set to 10, otherwise the proportional band will be set to 50.

```
IF VAR1 < 3 THEN CONPROP( 1 , 10 ) ELSE CONPROP( 1 , 50 )
```

Related topics

- *CONBIAS* on page 263
- *CONRATE* on page 264
- *CONRESET* on page 265

CONRATE

Used to change the rate (derivative component) of a PID control loop. *PIDcontroller* is any controller in the network. *Expression* is the new rate to set within the range of 0–2.00. The value for *Expression* can also be a variable or a computation.

Note: For KMD controllers only. For BACnet controllers, see the topic *Mnemonics for BACnet controllers* on page 227.

Syntax: *CONRATE(_PIDcontroller_,_Expression_)*

```
IF VAR1 < 3 THEN CONRATE( 1 , 1.3 ) ELSE CONRATE( 1 , 0.7
)
```

Related topics

- *CONBIAS* on page 263
- *CONPROP* on page 264
- *CONRESET* on page 265

CONRESET

Use CONRESET to change the reset rate (integral term) of a PID control loop in a KMD controller. *Controller#* is any PID control loop in the network. *Expression* is the number of allowable resets per hour within the range of 0–255. The value for *Expression* can also be a variable or a computation.

Note: KMD controllers only. For BACnet controllers, see the topic *Mnemonics for BACnet controllers* on page 227.

Syntax: *CONRESET(_controller#_,_Expression_)*

```
IF VAR1 < 3 THEN CONRESET( 1 , 25 ) ELSE CONRESET( 1 , 20
)
```

Reset for controller #1 is 20 unless variable VAR1 is less than 3; then *reset* will be set to 25.

Related topics

- *CONBIAS* on page 263
- *CONPROP* on page 264
- *CONRATE* on page 264

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.

Rules for using constants:

- 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 288
- *LOCALS* on page 292
- *REAL* on page 314
- *STRING* on page 327

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)
```

KMD example

```
A = COS( VAR10 )
```

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 257.

```
A = COS-1( VAR10 )
```

COV

The COV command sets up an off-panel object 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, lifetime, confirmed, increment)*

Note: This applies only to controllers with Generation 5 Control Basic.

- Within a controller, execute the COV command only once for each off-panel object.
- Add *NETPOINTCACHE* to every program in the controller that uses the off-panel object that is referenced by the COV command.

- Combine COV with *NETPOINT* to periodically update the off-panel value regardless of change.
- Use *ERRORLEVEL* to detect when a COV subscription is not returning a valid value from the off-panel point.

Table 29–8 COV 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 *Mnemonics for BACnet controllers* on page 227.
property	Present value (PV) is typically the property.	Priority is ignored for read only objects such as inputs.
lifetime	The time, in seconds, for the lifetime of the COV subscription.	
confirmed	Enter 0 (zero) for a confirmed subscription and 1 (one) for an unconfirmed subscription.	
increment	The increment for the change of value that the subscribed object will report.	

In the following example, the COV statement sets up a subscription to Binary Value object BV1 in controller Device Instance 5. It also sets up NETPOINT to read the value of BV1 every 300 seconds (5 minutes). The statements with COV and NETPOINT run only during the first scan after a restart.

Error detection is added with ERRORLEVEL. When the program executes the first time and there is reference to off-panel object BV1, the value is returned as *NAN* (Not A Number). The value may continue to be returned as NAN until a valid subscription is established. In addition, NAN is returned if the program requests the off-panel value and after three attempts does not receive a reply. When a valid number is returned, ERRORLEVEL is *FALSE*(0) and the COV value for the off-panel BV1 is used.

```
NETPOINTCACHE(1)
IF POWERLOSS THEN
    NETPOINT(5.BV1.PV,300,NAN)
    COV(5.BV1.PV,300,0,1)
ENDIF

IF 5.BV1=1 AND ERRORLEVEL=0 THEN START BV1 ELSE BV1=0
END
```

To use 5.BV1 in other programs within the same controller, add only the `NETPOINTCACHE(1)` command to the program.

```
NETPOINTCACHE(1)
IF 5.BV1=1 AND ERRORLEVEL=0 THEN START BV1 ELSE BV1=0
END
```

Related topics

- *ERRORLEVEL* on page 278
- *NAN* on page 296
- *NETPOINT* on page 297
- *NETPOINTCACHE* on page 298
- *POWERLOSS* on page 311
- *Transferring values between BACnet controllers* on page 232

DALARM

The DALARM statement initiates an alarm notification from a KMD controller. It may also initiate a dialing sequence in a modem connected directly to the controller in which the alarm notification is created.

Syntax: *DALARM_expression_,_delay_,string*



Note: For KMDigital controllers only.

Expression The *expression* parameter defines the condition that triggers the alarm. It is expressed as a conditional such as X_<_Y or X_>_Y or X_=_Y. The values for X and Y can be any number, variable, input, or output in any controller. Spaces within the expression are required and are indicated by the underscore (_).

Delay The value for *delay* is time expressed in whole seconds. Using *delay* minimizes multiple alarms when conditions change rapidly between alarm conditions. The value for *delay* may be up to 32,000 seconds.

String This is the text message that is added to the alarm viewer list. The message may be up to 69 characters long.

The following example sets up an alarm condition if the Input *IN1* is greater than 72 degrees for 7 seconds or longer. The delay parameter eliminates unnecessary alarms if input *IN1* briefly exceeds 72 degrees.

```
DALARM IN1 > 72 , 7 , Room temperature is above 72
degrees!
```

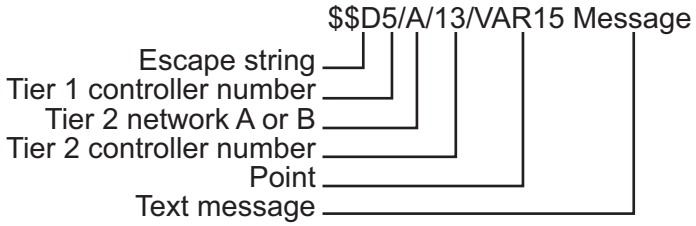
Use the following alarm message for TotalControl alarms. The text after *$$D* indicates the address of the controller that generated the alarm. This is explained in detail under *Setting up routing for TotalControl alarms* on page 269.

```
DALARM IN1 > 72 , 7 , $$D5/A/13/IN1 Room temperature is
above 72
```

Setting up routing for TotalControl alarms

Without TotalControl alarm routing, the text defined by *string* will be added to the alarm viewer but not the point of the alarms origin. To use TotalControl alarm routing, you must add to *string* information about the point that originated the alarm. The format is shown in the illustration *Routing message string for DALARM* on page 269.

Illustration 29–2 Routing message string for DALARM




```
DALARM IN1 > 72 , 7 , $$D5/A/13/VAR15 Room temperature is
above 72 degrees!
```

For WinControl XL email routing

To duplicate the WinControl XL Plus email feature, create one or more alarm classes named *E1-E8*.

Add the alarm class to the alarm message string as shown in the following example:

```
ALARM IN4 > 85 , 3 , E1 Area temp too high
```

For pager notification

By adding *CALL*, *NPAGE*, or *TPAGE* options to the text string, the controller initiates a dialing sequence to either pagers or a computer running WinControl XL. The KMD Tier 1 controller that initiates the alarm must have a modem connected to it for pager notification.

Table 29–9 Pager message string options

Alarm device	String option
Numeric pager	NPAGE(_1234567_)_5555555
Text pager	TPAGE(_1234567-1234_)_Message

```
DALARM IN1 > 72 , 7 , NPAGE( 1234567 ) 5555555
DALARM IN1 > 72 , 7 , TPAGE( 1234567 1234 ) Message
```

Related keywords

- *ALARM* on page 252
- *NPAGE* on page 301
- *TPAGE* on page 335

- *PHONE* on page 309

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 286.

Standard BACnet Control Basic example

```
10 DEC( AV1 , A + B )
20 DEC( AV2 )
```

Next Generation Control Basic example

```
DEC( AV1 , A + B )
DEC( AV2 )
```

KMD example

```
DEC( VAR1 , A + B )
DEC( VAR2 )
```

DECOM

Returns *true* if the decommission flag for a point is set and *false* if not.

Syntax: *DECOM(_point_)*

Note: KMD Tier 2 controllers only.

```
A = DECOM( IN1 )
```

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_)*

Note: For KMD controllers, see *DEW-POINT* on page 271.

See the related topic *DEWPOINTSI* on page 271 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: Deprecated for BACnet controllers. See the keyword *DEWPOINT* on page 270.

```
D = DEW-POINT( VAR1 , VAR2 )
```

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 270 to express temperature in degrees Fahrenheit.

Note: For KMD controllers see *DEW-POINT-SI* on page 271.

```
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 271.

```
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
```

KMD example

```
DISABLE OUT1
DISABLE A
DISABLE PRG1
```

Related topics

- *ENABLE* on page 275
- *START* on page 325
- *STOP* on page 326

DOM

Returns the current day of the month.

Tip: For the most accurate results, use DOM only once in a program and then assign the value to a local variable.

```
D = DOM
IF D = 5 THEN START X ELSE STOP X
```

DOW

A function that returns a numerical value for the day of the week.

Tip: For the most accurate results, use DOW only once in a program and then assign the value to a local variable.

```
D = DOW
IF D = MON THEN START X ELSE STOP X
```

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.



Tip: For the most accurate results, use DOY only once in a program and then assign the value to a local variable.

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.

```
D = DOY
IF D = DEC 25 THEN START X ELSE STOP X
```

EMAILD

Sends an email message with the value of system points as the contents of the message. Data from up to eight points may be included in one message.



Note: KMDigital web enabled products only.

Syntax: *EMAILD_to address_,_reply address_,_subject_,_data points*

In the following example the 15 second WAIT statement at line 20 halts program execution until the message is sent.

```
EMAILD joedoe@anonymous.com , webLite@jobsite.com , Air
Handler Data , IN1 VAR2 OUT2
WAIT 0:00:15 : REM Line 20
```

Related topics

- *EMAILL* on page 274
- *EMAILM* on page 274
- *EMAILR* on page 275

EMAILL

Sends an email message with data from either a trend log or runtime log as the contents of the message. Only one log may be sent in the same message. The log data in the body of the message is separated by commas (,).



Note: KMDigital web enabled products only.

Syntax: *EMAILL_to address_,_reply address_,_subject_,_log*

In the following example, the 15 second WAIT statement at line 20 halts program execution until the message is sent.

```
EMAILL joedoe@anonymous.com , webLite@jobsite.com , Roof
top unit , TL1
WAIT 0:00:15 : REM Line 20
```

Related topics

- *EMAILD* on page 273
- *EMAILM* on page 274
- *EMAILR* on page 275

EMAILM

Sends predefined text as the email message contents.



Note: KMDigital web enabled products only.

Syntax: *EMAILM_to address_,_reply address_,_subject_,_text*

In the following example, the 15 second WAIT statement at line 20 halts program execution until the message is sent.

```
EMAILM joedoe@anonymous.com , WebLite@jobsite.com ,
Refrigeration Alarm , Temperature in cooler is high!
WAIT 0:00:15 : REM Line 20
```

Related topics

- *EMAILD* on page 273
- *EMAILL* on page 274
- *EMAILR* on page 275

EMAILR

Sends an email message with data from either a trend log or runtime log as the contents of the message. Only one log may be sent in the same message. The log data in the body of the message is formatted in columns.

Note: KMDigital web enabled products only.

Syntax: *EMAILR_to address_,_reply address_,_subject_,_log*

In the following example, the 15 second WAIT statement at line 20 halts program execution until the message is sent.

```
EMAILR joedoe@anonymous.com , WebLite@jobsite.com , Roof
top unit , TL1
WAIT 0:00:15 : REM Line 20
```

Related topics

- *EMAILD* on page 273
- *EMAILL* on page 274
- *EMAILM* on page 274

ENABLE

ENABLE sets the value of an object, which can be the present value of an input, output, or value object, input point, output point, variable or program to 1 or *on*.

Syntax: *ENABLE_point*

BACnet example

```
ENABLE AO1
ENABLE A
```

KMD example

```
ENABLE OUT1
ENABLE A
ENABLE PRG1
```

Related topics

- *DISABLE* on page 271
- *START* on page 325
- *STOP* on page 326

END

Terminates the execution of a program. When the END statement is encountered, the controller stops reading lines and exits the program. All program lines that follow an encountered END statement are *not* executed.

In the following examples, 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: This applies 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 280 and *User-defined functions and procedures* on page 244.

ENDPROC

Use ENDPROC to mark the end of the definition of a Control Basic user defined procedure.



Note: This applies only to controllers with Generation 5 Control Basic.

```
PROCEDURE lowOAT(a,b)
  AV1 = MIN(a,b)
ENDPROC
```

See also the topics *PROCEDURE* on page 312 and *User-defined functions and procedures* on page 244.

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 277 to enter OAT in degrees Celsius.

For KMD controllers, see the topic *ENTHALPY-SI* on page 278 to enter OAT in degrees Celsius.

BACnet example

```
E = ENTHALPY( AI1 , AI2 )
```

KMD example

```
E = ENTHALPY( VAR10 , VAR20 )
OAE = ENTHALPY-SI( OAH, OAT )
```

ENTHALPYSI

This function 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_)*



Note: For KMD controllers see *ENTHALPY-SI* on page 278.

See the topic *ENTHALPY* on page 277 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

This function 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 277.

```
E = ENTHALPY-SI( VAR10 , VAR20 )
E = ENTHALPY-SI( OAH , OAT )
```

ERROR-COUNT

Returns the number of network errors detected by the controller since the last start-up cycle or executed a CLEAR-COUNT statement.



Note: KMDigital Tier 1 controllers only.

```
VAR1 = ERROR-COUNT
```

The following statements return a count of the errors detected for a specific network on connections in Tier 1 controllers:

```
ERROR-COUNT-SUBA
ERROR-COUNT-SUBB
ERROR-COUNT-MAIN
ERROR-COUNT-ETHERNET
ERROR-COUNT-PCA
ERROR-COUNT-PCB
```

See the related topic *CLEAR-COUNT* on page 262.

ERRORLEVEL

The ERRORLEVEL function returns a non zero (0) value when an error is detected in the previous statement.

Syntax: *ERRORLEVEL*



Note: This keyword is for Next Gen and Gen 5 controllers only.

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 254.

FOR TO NEXT

The FOR TO NEXT loop repeats a set of instructions a specific number of times.

Syntax: *FOR_ControlVariable_=_StartValue_to_EndValue(_Step_Increment_)*

- *ControlVariable* is the variable that *FOR* increments each time the loop repeats. It controls whether 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 it is, 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* or *CON1* 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
```

KMD example

```
FOR A = 0 TO CON1 STEP 0.1
OUT1 = A
WAIT 0:00:10
NEXT A
```

FRETURN

Use FRETURN to designate the value that is returned by the function. A function can include multiple FRETURN statements, bu, the result is returned from only the first FRETURN executed.



Note: This applies 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 280 and *User-defined functions and procedures* on page 244.

FUNCTION

Use FUNCTION to mark the beginning of a user-defined function. A function is a self-contained block of statements that returns a single value.



Note: This applies 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 functions and procedures* on page 244
- *FRETURN* on page 280
- *ENDFUNC* on page 276
- *PROCEDURE* on page 312

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 it encounters a RETURN statement. 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 282 and *RETURN* on page 315.

BACnet example

```
GOSUB 30
END
REM Line 30 subroutine starts here
RETURN
```

Next generation Control Basic

```
     GOSUB DoSubRoutine
     END
DoSubRoutine:
     RETURN
```

KMD example

```
GOSUB 30
END
REM Line 30 subroutine start here
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 281.

Syntax: *GOTO_line#*

BACnet example

```
GOTO 30
START BO1
REM Line 30 Program continues here
END
```

Next generation Control Basic

```
     GOTO JumpToEnd
     START BO1
JumpToEnd:
     END
```

KMD example

```
GOTO 30
START OUT1
REM Line 30
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"
```

HANGUP

Use HANGUP to end a telephone call placed through a modem connected to the KMD network. When Control Basic runs this statement, the controller drops the modem off-line to end the call.

Note: For KMDigital controllers only.

This example program will dial the number and after connecting, will wait 10 seconds before hanging up.

```
PHONE ATDT 555-1234
IF INTERVAL( 0:00:10 ) THEN 30 ELSE END
HANGUP : REM Line 30
```

See the related keyword topic *PHONE* on page 309.

HAVE-TOKEN

Returns *true* if the controller has the network token and *false* if not.

Note: KMD Tier 2 controllers only.

```
IF HAVE-TOKEN THEN START OUT1
```

HSEL

This function elects 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)
```

KMD example

```
A = HSEL( 2 , IN1 , IN2 , IN3 , VAR1 )
```

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 AO5 if Analog Input AI1 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
```

KMD example

```
IF IN1 < IN2 THEN STOP OUT5 ELSE START OUT5
END
```

In this example, the program stops output OUT5 if input IN1 is less than input IN2. If IN1 is not less than IN2, OUT5 will be turned on (started). If the *ELSE START OUT5* statement was not included, the program will stop OUT5 if IN1 is less than IN2. Otherwise, it will do nothing and end the program.



Note: Use commas to separate multiple commands in an IF statement.

```
IF IN1 > VAR1 THEN START OUT2 , STOP OUT1
IF IN1 > VAR1 THEN START OUT2 , STOP OUT1 , OUT3 = 0 ELSE
STOP OUT1 , OUT3 = CON1
```

IF+ THEN

IF+ THEN 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 or label ADD1, 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 284 and *IF- THEN* on page 286.

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 ADD1
END
ADD1:
AV13 = AV13 + 1
RETURN
```

KMD example

```
IF+ SENSOR-ON( IN1 ) THEN GOSUB 30
END
VAR13 = VAR13 + 1 : REM Line 30 start here
RETURN
```

IF- THEN

IF- THEN 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 284 and *IF+ THEN* on page 285.

INC

Use INC to increment 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 270.

Standard BACnet Control Basic example

```
INC( AV1 , A + B )
INC( AV2 )
```

Next Generation Control Basic example

```
INC( AV1 , A + B )
Inc( AV2 )
```

KMD example

```
INC( VAR1 , A + B )
INC( VAR2 )
```

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 or variable VaR1.

Standard BACnet Control Basic example

```
AV1 = INT( TIME / 3600 )
```

Next Generation Control Basic example

```
AV1=INT(TIME/3600)
```

KMD example

```
VAR1 = INT( TIME / 100 )
```

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, *BACnet data types* on page 247.

Note: This applies only to controllers with Generation 5 Control Basic.

Syntax: *INTEGER variable[, variable, ...]*

Rules for the INTEGER command:

- 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]*

Rules for the INTEGER command in an array:

- 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 292
- *REAL* on page 314
- *STRING* on page 327
- *Programming with variables* on page 226
- *BACnet arrays and dynamic access* on page 246

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 either Analog Value Object AV1 or variable VAR2—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
```

KMD example

```
IF INTERVAL( 00:00:45 ) THEN VAR2 = VAR2 + .1
END
```

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 291.

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 AI4 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
```

ISSTRING

ISSTRING tests the value of *expression* to determine if it is of data type character string.

Syntax: *ISSTRING(expression)*



Note: This applies only to controllers with Generation 5 Control Basic.

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 289.

```
B = LN-1( IN4 * 125 )
```

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 )
```

KMD example

```
A = LN( IN1 )
```

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
```

Rules for LOCALS declaration:

- 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 288
- *REAL* on page 314
- *STRING* on page 327
- *Programming with variables* on page 226
- *BACnet arrays and dynamic access* on page 246

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)
```

KMD example

```
A = LSEL( 2 , IN1 , IN2 , IN3 , VAR1 )
```

MAX

MAX returns the maximum value of the expression listed. Expressions can be one of the following items.

- Input, output, or value objects
- Input, output, or variable points
- 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)
```

KMD example

```
A = MAX( IN1 , IN2 , IN3 , VAR1 )
```

MIN

MIN returns the minimum value of those expression listed. Expressions can be one of the following items.

- Input, output, or value objects
- Input, output, or variable points
- 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 )
```

KMD example

```
B = MIN( IN1 , IN2 , IN3 , VAR1 )
```

MOD

MOD is an arithmetic operator function that returns the remainder (or modulus) 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
```

KMD example

```
IF VAR1 MOD 5 = 0 THEN START OUT1 ELSE STOP OUT1
```

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 225.

MODBUSTRANSFER

Use MODBUSTRANSFER to control the transfer of data between pairs of points and registers set up with interprotocol points.

Syntax: *MODBUSTRANSFER(point , action)*

- *Point* designates the interprotocol point.
- *Action* controls the transfer. See the following table for a list of permissible actions.

Note: Only for KMDigital models with a Modbus license.

```
MODBUSTRANSFER( 1 , DISABLED )
MODBUSTRANSFER( 2 , ONCE )
MODBUSTRANSFER( 3 , ALWAYS )
```

Table 29–10 Actions

Action	Description
DISABLED	Disables the transfer of data from the read point to the write point.
ONCE	Initiates a one-time transfer of data between the read point and the write point. When the transfer is complete, the mode of the point pair is set to DISABLED.
ALWAYS	Initiates a continuous transfer of data between the read point and the write point.

MODELNUMBER

MODELNUMBER returns the numerical portion of the model number of the controller.

Syntax: *MODELNUMBER*

Note: For KMD controllers see *MODEL-NUMBER* on page 295.

```
AV1=MODELNUMBER
```

MODEL-NUMBER

MODEL-NUMBER returns the numerical portion of the model number of the controller.

Note: Deprecated for BACnet controllers. See the keyword *MODELNUMBER* on page 295.

```
VAR1 = MODEL-NUMBER
```

MONTH

MONTH returns the current month of the year.

Standard BACnet Control Basic example

```
M = MONTH
```

Next Generation Control Basic example

```
M=MONTH
```

KMD 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 289.

NEG_INFINITY

Use the NEG-INFINITY command to set a variable or property to a very large negative number.



Note: Applies only to controllers with Next Generation or 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 310.

NETPOINT

Use the NETPOINT command to set the interval for reading from and writing to BACnet off-panel objects. NETPOINT and COV are the preferred methods to read from or write to off-panel points in Generation 5 controllers.

Syntax: *NETPOINT(device instance.object.property, read interval, write interval)*

Note: This applies only to controllers with Generation 5 Control Basic.

Rules for the NETPOINT command:

- Within a controller, execute the NETPOINT command only once for each off-panel object.
- Add *NETPOINTCACHE* to every program in the controller that uses the off-panel object that is referenced by the NETPOINT command.
- Only the shortest read or write interval for the same object in all NETPOINT statements within the controller will be used.

NETPOINT parameters

Parameter	Description	Comments
device	The device instance number.	
object	A valid Control Basic object mnemonic.	See *Mnemonics for BACnet controllers* on page 227.
property	The property for reading or writing.	Present Value is the default property.
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.

In the following example, the statement with NETPOINT runs only during the first scan after a restart. The NETPOINT function sets up reading Analog Value object AV1 in controller Device Instance 5 every 180 seconds (3 minutes) and writing to AV1 every 60 seconds. Then, local variable *A* is updated every 180 seconds (3 minutes) and the value of local variable *B* writes to AV1 in device 5 every 60 seconds.

```
NETPOINTCACHE(1)
IF POWERLOSS THEN
    NETPOINT(5.AV1.PV,180,60)
ENDIF
A=5.AV1
5.AV1@8=B
```

To use 5.AV1 in other programs within the same controller, add only the `NETPOINTCACHE(1)` command to the program.

```
NETPOINTCACHE(1)
IF 5.AV1 < 51 THEN STOP BV1
```

See also the topics *POWERLOSS* on page 311 and *NETPOINTCACHE* on page 298.

NETPOINTCACHE

Use NETPOINTCACHE to enable a cache of values within a program from off-panel objects that are referenced with the *COV* and *NETPOINT* functions. Using NETPOINTCACHE with NETPOINT or COV reduces network traffic by sharing values among all programs within the same controller. This is the preferred method to read from or write to off-panel points in Generation 5 controllers.

Syntax: *NETPOINTCACHE(expression)*



Note: This applies only to controllers with Generation 5 Control Basic.

Rules for the NETPOINTCACHE command:

- Add NETPOINTCACHE to every program in a controller that is referencing an off-panel object by using NETPOINT or COV.
- Enter one (1) for *expression* to enable NETPOINTCACHE and COV caching.
- Enter zero (0) for *expression* to disable NETPOINTCACHE and COV caching. This is the default state in controllers that support Generation 5 Control Basic.

In the following example, the statement with NETPOINT runs only during the first scan after a restart. The NETPOINT function sets up reading Analog Value object AV1 in controller Device Instance 5 every 180 seconds (3 minutes) and writing to AV1 every 60 seconds. Then, local variable *A* is updated every 180 seconds (3 minutes) and the value of local variable *B* writes to AV1 in device 5 every 60 seconds.

```
NETPOINTCACHE(1)
IF POWERLOSS=0 THEN
    NETPOINT(5.AV1.PV,180,60)
ENDIF
A=5.AV1
5.AV1@8=B
```

To use 5.AV1 in other programs within the same controller, add `NETPOINTCACHE(1)` to the program.

```
NETPOINTCACHE(1)
IF 5.AV1 < 51 THEN STOP BV1
```

Related topics

- *COV* on page 266
- *NETPOINT* on page 297
- *POWERLOSS* on page 311
- *Transferring values between BACnet controllers* on page 232

NETSENSORSTATUS

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.

Note: Deprecated for BACnet controllers. See the keywords IF THEN (ELSE).

Note: For KMD controllers see *NETSENSOR-STATUS* on page 300.

NETSENSORSTATUS can be replaced with the following command line in Generation 5 and Next Generation Control Basic.

```
IF DS(1).KMC_STAT_STATUS=1 THEN U=1 ELSE=0
```

where the value of U is used in place of the NETSENSORSTATUS command.

NETSENSORSTATUS can also replaced with the following command line using the EXISTS command in conjunction with an IF-THEN-ELSE command in Generation 5 and Next Generation Control Basic.

```
IF EXISTS(DS1.KMC_STAT_STATUS) THEN U = DS1.KMC_STAT_STATUS
ELSE U = 0
```

Standard Control Basic example

```
F NOT NETSENSORSTATUS THEN STOP BV1
```

NETSENSOR-STATUS

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: Deprecated for BACnet controllers. See the keyword *NETSENSORSTATUS* on page 300.

```
DALARM NETSENSOR-STATUS = 0 , 10 , NetSensor in lobby not
responding!
```

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
10 IF NOT OUT1 THEN STOP OUT2
```

See the related topic *Using Boolean logic* on page 225.

NPAGE

Use the NPAGE command to initiate a telephone call to a numeric pager.

Syntax: *NPAGE telephone number , numeric message*



Note: For KMDigital controllers only.

- Use NPAGE only in programs running in Tier 1 controllers that are connected to a modem.
- *Telephone number* cannot contain spaces, hyphens, or punctuation other than commas (,). Each comma adds a two second pause.

```
DALARM VAR51 , 1 , VAR 51 is in alarm!
ON-ALARM 40
GOTO 60
NPAGE 18005551212,,,,51 : REM line 40
WAIT 0:00:15
END : Rem Line 60
```

See the related topics *TPAGE* on page 335 and *PHONE* on page 309.

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. For example, if *expression* equals 2 the program will branch to the second location in the list; if *expression* equals 5 the program will branch to the fifth location in the list. *Expression* is rounded to an integer. 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 315.

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 : REM Line 40
RETURN : REM Line 40
RETURN : REM Line 40
```

Next Generation Control Basic

In this example, Value Object *AV1* is equal to *3*, which will cause the program to branch to label *Third*. If *AV1* equals *2*, the program will branch to label *Second*, etc.

```
AV1 = 3
ON AV1 GOSUB First , Second , Third
END
First:
RETURN
Second:
RETURN
Third:
RETURN
```

KMD example

In this example, Variable *VAR1* is equal to *3*, which will cause the program to branch to Line *80*. If *VAR1* equals *2*, the program will branch to Line *60*, etc.

```
VAR1 = 3
ON VAR1 GOSUB 40 , 50 , 60
END
RETURN : REM Line 40
RETURN : REM Line 50
RETURN : REM Line 60
```

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. For example, if *expression* equals 2, the program will branch to the second location in the list; if *expression* equals 5, the program will branch to the fifth location in the list. *Expression* is rounded to an integer. 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 Line 40 Program continues here
REM Line 50 Program continues here
REM Line 60 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:
```

KMD example

In this example, VAR1 is equal to *3*, which will cause the program to branch to line 60. If VAR1 were equal to 2, the program would branch to line 50, etc.

```
VAR1 = 3
ON VAR1 GOTO 40 , 50 , 60
END
REM Line 40 Program continues here
REM Line 50 Program continues here
REM Line 60 Program continues here
```

ON-ALARM

ON-ALARM is a control statement that redirects the program when alarms are detected in a KMD controller. When an alarm condition is detected, ON-ALARM redirects the program to the line number listed. This redirection occurs only once when the alarm is first detected. This statement only evaluates the ALARM statement immediately prior to it.

Syntax: *ON-ALARM_line#*



Note: KMD controllers only.

The following program runs the first three lines if alarms are not detected. However, if the alarm is activated on the first line, when ON-ALARM runs, the program jumps to the fourth line (Line 40). In this example, if no alarm is detected the program does nothing. If an alarm is detected Output *OUT1* is turned on.

```
ALARM IN1 < 34 , 1 , Temp is near freezing
ON-ALARM 40
END
START OUT1 : REM Line 40-Activates warning device
END
```

ONERROR

ONERROR is a control statement that redirects the program to the line specified by *location* when an error is detected in the previous line.

Syntax: *ONERROR location*



Note: For KMD Controllers see *ON-ERROR* on page 305.

In the following examples, the program attempts to read an off-panel object. If the object is not found, the program substitutes the value *55*.

Standard BACnet Control Basic example

```
IF INTERVAL ( 00:01:00 ) THEN AV24 = 4410.AI4
ONERROR 40
GOTO 60 : REM Jump around error recovery
REM Line 40-Error recovery
AV16 = 55
REM Line 60-Continue program
```

Next Generation Control Basic

```
IF INTERVAL ( 00:01:00 ) THEN
    REM Verify that the value is good
    IF ISNAN( 4410.AI4 ) THEN
        REM Set a default value
        AV24 = 55
    ELSE
        REM Use the received value
        AV24 = 4410.AI4
    ENDIF
ENDIF
```

ON-ERROR

ON-ERROR is a control statement that redirects the program to the line specified by *location* when an error is detected in the previous line.

Syntax: *ON-ERROR location*



Note: Deprecated for BACnet controllers. See the keyword *ONERROR* on page 305.

The example prints an error message if line 10 is unsuccessful while executing the *PHONE* statement. *ON-ERROR* may also be used to branch to a subroutine where a second telephone phone number is called when an error may be received on the first attempt. Use *ON-ERROR* with the following commands.

- BAC-GET
- BAC-SET
- BAC-RLQ
- NPAGE
- PHONE
- REMOTE-GET
- REMOTE-SET
- RUN-MACRO
- TPAGE

```
PHONE 555-1234
ON-ERROR 40
END
REM Line 40 handles the error
END
```

OPEN

The OPEN command sets 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
```

KMD example

```
OPEN OUT1
OPEN A
OPEN VAR1
```

Related topics

- *CLOSE* on page 263
- *START* on page 325
- *STOP* on page 326

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 225.

OUTPUTOVERRIDE

The OUTPUTOVERRIDE function 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)*



Note: For BACnet controllers only. For KMD controllers see the keyword *OUTPUT-OVERRIDE*.

The function returns *FALSE* (0) if the switch is set to A (Auto) and *TRUE* (1) if the switch is set to either the O (Off) or H (Hand controlled On) position. The argument *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

The OUTPUT-OVERRIDE function 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(_output_)*



Note: Deprecated for BACnet controllers. See the keyword *OUTPUTOVERRIDE* on page 307.

The function returns *FALSE* (0) if the switch is set to A (Auto) and *TRUE* (1) if the switch is set to either the O (Off) or H (Hand controlled On) position. The argument *output* can be expressed as either of the following:

- The point number of the output.
- A local variable whose value represents the number of an output object.

```
DALARM OUTPUT-OVERRIDE( 1 ) , 300 , OUTPUT1 1 SWITCH IS
NOT IN AUTO
IF OUTPUT-OVERRIDE( 2 ) THEN STOP OUT2
```

PANELADDRESS

PANELADDRESS returns the device instance number of the controller on which the Control Basic program is running.



Note: For KMD controllers see *PANEL-ADDRESS* on page 308.

Standard Control Basic example

```
P = PANELADDRESS
```

Next Generation Control Basic example

```
P=PANELADDRESS
```

PANEL-ADDRESS

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 308

```
P = PANEL-ADDRESS
```

PHONE

Use PHONE to establish a modem connection from a controller to a remote computer running WinControlXL Plus. Use PHONE in Control Basic programs only in Tier 1 controllers with an attached modem.

Syntax: *PHONE_ATDT_telephone number*

- The *ATDT* option is a standard modem structure clause to configure the modem to dial out in a tone format.
- *Telephone number* cannot contain spaces, hyphens or punctuation other than commas (,). Each comma adds a two second pause.

When Control Basic runs the *Phone* statement, the processor suspends operation for up to 45 seconds as it waits for the modem to return a connection string.



Caution: The WAIT statements in the following examples are critical to reliable program execution. Do not delete them.

Example—Tier 1 controller

The sequence in the following example dials the number, waits up to 45 seconds for the remote modem to answer, and then after connecting to it, waits 1 minute before hanging up. The HANGUP command is required at the end of the sequence.

```
DALARM VAR51 , 1 , VAR 51 is in alarm!
ON-ALARM 40
GOTO 80
PHONE 18005551212,,,,51
WAIT 0:00:45 : ON-ERROR 40
HANGUP
WAIT 0:01:00
END
10 DALARM 1-VAR51 , 1 , VAR 51 is in alarm!
20 ON-ALARM 40
30 GOTO 80
40 PHONE 18005551212,,,,51
50 WAIT 0:00:45 : ON-ERROR 40
60 HANGUP
70 WAIT 0:01:00
80 END
```

Example–Tier 2 controller

In the following example, a Tier 2 controller dials a pager and leaves the message 1234. Each comma adds a 1 second pause.

```
10 IF+ IN1 THEN 20 ELSE END
20 PHONE ATDT 5551212,,,,,,1234
30 HANGUP
```

Related topics

- *NPAGE* on page 301
- *TPAGE* on page 335
- *HANGUP* on page 283

PI

This function PI inserts the mathematical value of pi(π). 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)
```

KMC 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 or Generation 5 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 296.

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*.

Note: For KMD controllers see *POWER-LOSS* on page 311.

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 311.

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: This applies only to controllers with Generation 5 Control Basic.

See also, *PROCEDURE* on page 312 and *User-defined functions and procedures* on page 244.

PROCEDURE

Use PROCEDURE to mark the beginning of a user-defined procedure. A procedure is a self-contained block of Control Basic statements that performs a task.

Note: This applies 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 parentheses ( ) 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 functions and procedures* on page 244
- *PRETURN* on page 311
- *FUNCTION* on page 280

READ-CONBIAS

READ-CONBIAS returns the bias value of a PID control loop. The value range is 0–100.

Syntax: *READ-CONBIAS(_expression_)*

Note: KMD controllers only.

```
B = READ-CONBIAS( 1 )
```

READ-CONPROP

READ-CONPROP returns the proportional band value of PID control loop. The value range is 0–4000.

Syntax: *READ-CONPROP(_expression_)*



Note: KMD controllers only.

```
P = READ-CONPROP( 1 )
```

READ-CONRATE

READ-CONRATE returns the rate (derivative) value of PID controller. The value range is 0–2.00.

Syntax: *READ-CONRATE(_expression_)*



Note: KMD controllers only.

```
10 R = READ-CONRATE( 1 )
```

READ-CONRESET

READ-CONRESET returns the reset (integral) value of a PID control loop. The value range is 0–255.

Syntax: *READ-CONRESET(_expression_)*



Note: KMD controllers only.

```
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, *BACnet data types* on page 247.

Note: This applies only to controllers with Generation 5 Control Basic.

Syntax: *REAL variable[, variable, ...]*

Rules for the REAL command:

- 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 288
- *LOCALS* on page 292
- *STRING* on page 327
- *Programming with variables* on page 226.
- *BACnet arrays and dynamic access* on page 246

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
```

KMD example

```
REM ** Step temperature every minute by 1 degree **
IF INTERVAL( 0:01:00 ) THEN VAR1 = VAR1 + 1
REM **calculation for velocity (FPM)**
VAR1 = 4004.4 * SQR( IN1 )
END
```

RETURN

The RETURN command returns control from a subroutine that was called with a GOSUB or ON GOSUB statement.

See the related topics *GOSUB* on page 281 and *ON GOSUB* on page 302.

RLQ

Relinquishes the priority level of a BACnet object.

Note: For BACnet controllers only.

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 that 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
```

KMD example

```
IF TIME = 20:00:00 + RND( 10:00:00 ) THEN START OUT1
```

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 the program longer to process and the number of scans per second decreases.

One application for SCANS is to create a time-based counter. 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 you use the INTERVAL or WAIT statements, you are limited to a time division no smaller than one second.

See the related topic *About Control Basic scans* on page 218.

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
```

KMD example

```
A = 1 / SCANS
B = A + B : REM B Will inc. by 1 every sec.(based on scan
rate)
IF B > 10 THEN B = 0 : REM B counts 0-10 in 10 seconds
END
```

SCHEDOFF

Use the SCHEDOFF function to find when a schedule object will next set the present value of the reference object to *INACTIVE* or a value of zero (0). The schedule object must be within the same controller as the controller running the program. The value returned is for the current day, as maintained in the controller, of the weekly schedule in a schedule object.

Syntax: *SCHEDOFF(_schedule object #_,_time_)*

Rules for the SCHEDOFF command:

- The value for *time* may be specified with a fixed value, local variable, another object, or the keyword TIME. If TIME is used, the time maintained in the controller running the program is the time value used.
- A fixed value for *time* is entered in the 24-hour format (hh:mm:ss).
- The returned value is the difference—in seconds—between the value specified by *time* and the next scheduled time that the schedule's present value will change to *INACTIVE* (0).
- A returned value of 0 indicates that the schedule is already set to *INACTIVE*.
- A return of 86,400 indicates that there are no more scheduled *INACTIVE* times for the current day.



Note: For BACnet controllers only. See *WS-OFF* on page 337 for an equivalent Control Basic function in KMD Controllers.



Note: KMC Controls recommends that, because it is computationally intensive, Control Basic should not continuously run the SCHEDOFF function.

In the following example, the SCHEDOFFF functions are evaluated once every five minutes. For a schedule that changes to *OFF* at 6:00 P.M., the values for local variables A, C, and D will equal 3600. The value for variable E will change with the time of day as maintained in the controller.

Standard Control Basic example

```
B = 61200 : REM 5:00 P.M.
AV30 = B
IF INTERVAL( 0:05:00 ) THEN A = SCHEDOFF( 1 , B )
IF INTERVAL( 0:05:00 ) THEN C = SCHEDOFF( 1 , AV30 )
IF INTERVAL( 0:05:00 ) THEN D = SCHEDOFF( 1 , 61200 )
IF INTERVAL( 0:05:00 ) THEN E = SCHEDOFF( 1 , TIME )
```

Next Generation and Generation 5 Control Basic example

```
B = 61200 : REM 5:00 P.M.
AV30 = B
IF INTERVAL( 0:05:00 ) THEN A = SCHEDOFF( 1 , B )
IF INTERVAL( 0:05:00 ) THEN C = SCHEDOFF( 1 , AV30 )
IF INTERVAL( 0:05:00 ) THEN D = SCHEDOFF( 1 , 61200 )
IF INTERVAL( 0:05:00 ) THEN E = SCHEDOFF( 1 , TIME )
```

See *SCHEDON* on page 318 for calculating the time when a schedule becomes active.

SCHED-OFF

Deprecated for BACnet controllers. See *SCHEDOFF* on page 317.

SCHEDON

Use the SCHEDON function to find when a schedule object will next set the present value of the reference object to *ACTIVE* or a non-zero value. The schedule object must be within the same controller as the controller running the program. The value returned is for the current day, as maintained in the controller, of the weekly schedule in a schedule object.

Syntax: *SCHEDON(_schedule object #_,_time_)*

Rules for the SCHEDON function:

- The value for *time* may be specified with a fixed value, local variable, another object, or the keyword TIME. If TIME is used, the time maintained in the controller running the program is the time value used.
- A fixed value for *time* is entered in the 24-hour format (hh:mm:ss).
- The returned value is the difference—in seconds—between the value specified by *time* and the next scheduled time that the schedule's present value will change to *Active* or a non-zero(0) value.
- A returned value of 0 indicates that the schedule is already set to *ACTIVE* or a non-zero(0) value.
- A return of 86,400 indicates that there are no more *ACTIVE* or a non-zero(0) values scheduled for the current day.

Note: For BACnet controllers only. See *WS-ON* on page 338 for an equivalent Control Basic function in KMD controllers.

Note: KMC Controls recommends that, because it is computationally intensive, Control Basic should not continuously run the SCHEDON function.

In the following example, the SCHEDON functions are evaluated once every five minutes. For a schedule that changes to *ON* at 6:00 A.M., the values for local variables A, C, and D will equal 1800. The value for variable E will change with the time of day as maintained in the controller.

Standard Control Basic example

```
B = 19800 : REM 5:30 A.M.
AV30 = B
IF INTERVAL( 0:05:00 ) THEN A = SCHEDON( 1 , B )
IF INTERVAL( 0:05:00 ) THEN C = SCHEDON( 1 , AV30 )
IF INTERVAL( 0:05:00 ) THEN D = SCHEDON( 1 , 19800 )
IF INTERVAL( 0:05:00 ) THEN E = SCHEDON( 1 , TIME )
```

Next Generation and Generation 5 Control Basic example

```
B = 19800 : REM 5:30 A.M.
AV30 = B
IF INTERVAL( 0:05:00 ) THEN A = SCHEDON( 1 , B )
IF INTERVAL( 0:05:00 ) THEN C = SCHEDON( 1 , AV30 )
IF INTERVAL( 0:05:00 ) THEN D = SCHEDON( 1 , 19800 )
IF INTERVAL( 0:05:00 ) THEN E = SCHEDON( 1 , TIME )
```

See *SCHEDOFF* on page 317 for calculating the time when a schedule becomes inactive.

SCHED-ON

Deprecated for BACnet controllers. See the keyword *SCHEDON* on page 318.

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#_)*



Note: For BACnet controllers only. For KMD controllers see the topic *SENSOR-OFF* on page 320.

When SENSOROFF is used with *SENSORON* on page 321 and *IF THEN (ELSE)* on page 284, *IF+ THEN* on page 285, or *IF- THEN* on page 286, 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 29–11 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 320, 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 that 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. A typical application is a two-wire thermistor sensor with *Up* and *Down* pushbuttons. SENSOR-ON detects 0 volts (closed contact) while SENSOR-OFF detects 5 volts (open-circuit) condition.

Syntax: *SENSOR-OFF(_IN#_)*

Note: Deprecated in BACnet controllers. See the keyword *SENSOROFF* on page 319.

When SENSOR-OFF is used with *SENSOR-ON* on page 322 and *IF THEN (ELSE)* on page 2840, *IF+ THEN* on page 285, or *IF- THEN* on page 286, you can determine three separate conditions from one input:

- A temperature or other analog reading.
- A sensor with open contacts (Sensor-Off).
- A sensor with closed contacts (Sensor-On).

These statements 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 29–12 Input conditions for SENSOR-OFF in KMD Controllers

Input Voltage	Detected condition
0	Closed circuit
0.1	Temperature-55 degrees
4.9	Temperature-95 degrees
5.0 or greater	Open circuit

In the example table, under normal temperature conditions, the input voltage would never fall below 0.1 volts. When a sensor is shorted to ground, the input voltage will fall to zero, which is a condition SENSOR-ON 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 that will be detected by SENSOR-OFF.

See the keyword topic *DECOM* on page 270 for additional information about decommissioned inputs.

```
10 IF- SENSOR-OFF( IN1 ) THEN GOSUB 30
20 END
30 VAR13 = VAR13 + 1
40 RETURN
```

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#_)*

Note: For BACnet controllers only. For KMD controllers see *SENSOR-ON* on page 322.

When SENSORON is used with *SENSOROFF* on page 319 and *IF THEN (ELSE)* on page 284, *IF+ THEN* on page 285, or *IF- THEN* on page 286, 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 29–13 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 322, 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 that 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. 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: *SENSOR-ON(_IN#_)*

Note: Deprecated for BACnet controllers. See *SENSORON* on page 321.

When SENSOR-ON is used with *SENSOR-OFF* on page 320 and *IF THEN (ELSE)* on page 284, *IF+ THEN* on page 285, or *IF- THEN* on page 286, you can determine three separate conditions from one input:

- A temperature or other analog reading.
- A sensor with open contacts (Sensor-Off).
- A sensor with closed contacts (Sensor-On).

These statements 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 29–14 Example table for SENSOR-ON for KMD controllers

Input Voltage	Detected condition
0	Closed circuit
0.1	Temperature-55 degrees
4.9	Temperature-95 degrees
5.0	Open circuit

In the table, *Example table for SENSOR-ON for KMD controllers*, the input voltage under normal temperature conditions would never fall below 0.1 volts. When a sensor is shorted to ground, the input voltage will fall to zero, which is a condition that SENSOR-ON 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 that will be detected by SENSOR-OFF.

```
IF+ SENSOR-ON( IN1 ) THEN VAR11 = 02:00:00
```



Note: Use *SENSOR-ON* to detect momentary, unusual conditions such as a pressed button. The opened or closed contact condition should last no longer than two minutes or the functions will be disabled. After three minutes the inputs will change to a decommissioned status but the commands will still execute.

See the keyword topic *DECOM* on page 270 for additional information about decommissioned inputs.

SIN-1

The function SIN-1 returns the arcsine of the specified angle. The value of *angle* is expressed in radians.

Syntax: *SIN-1(_angle_)*



Note: Deprecated for BACnet controllers. See the keyword *ARCSIN* on page 257.

```
A = SIN-1( VAR1 )
```

SIN

The SIN function returns the sine of the specified angle. *The value of angle* is expressed in radians.

Syntax: *SIN(_angle_)*

Standard Control Basic example

```
A = SIN( AI1 )
```

Next Generation Control Basic example

```
A=SIN(AI1)
```

KMD example

```
A = SIN( VAR10 )
```

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)
```

KMD example

```
A = SQR( IN1 )
```

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
```

KMD example

```
START OUT1
START VAR5
START A
START PRG1
```

Related topics

- *STOP* on page 326
- *DISABLE* on page 271
- *ENABLE* on page 275

STATUS

This function returns the network status of a KMD controller. The status codes are listed in the table *Status conditions* on page 326.

Syntax: *STATUS(_expression_)*

Note: For KMDigital controllers only.

Table 29–15 Status conditions

Value	Status
0	Loss of communication; Not on-line
1	Controller is functioning properly
2	Controller has no program but is communicating

```
REM Take special action if controller is not functioning
IF STATUS( 2 ) = 0 THEN GOSUB 40
END
REM Line 40
START OUT1 : STOP OUT2 : RETURN
```

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
```

KMD example

```
STOP OUT1
STOP VAR10
STOP PRG7
STOP RT1
```

Related topics

- *START* on page 325
- *DISABLE* on page 271
- *ENABLE* on page 275

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, *BACnet data types* on page 247.

Note: This applies 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 data type string or to other variables of data type string.
- Strings can be 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]*

Rules for using the STRING command to declare declare an array:

- 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 288
- *LOCALS* on page 292
- *REAL* on page 314
- *Programming with variables* on page 226
- *BACnet arrays and dynamic access* on page 246

TAN-1

The TAN-1 function returns 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 258.

```
A = TAN-1( VAR12 )
```

TAN

The TAN function 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)
```

KMD example

```
A = TAN( VAR10 )
```

TBL

Use the TBL function to calculate the 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_)*

Rules for using the TBL function:

- *Expression* is an analog input, value object, or variable.
- The value for *table* must be a whole number.
- In BACnet controllers, use a Control Basic (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 )
```

KMD example

```
VAR1 = TBL( IN3 , 2 )
VAR2 = TBL( VAR7 , 1 )
```

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

1 Enter values into Cbasic 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 29–3 Graph for boiler reset function

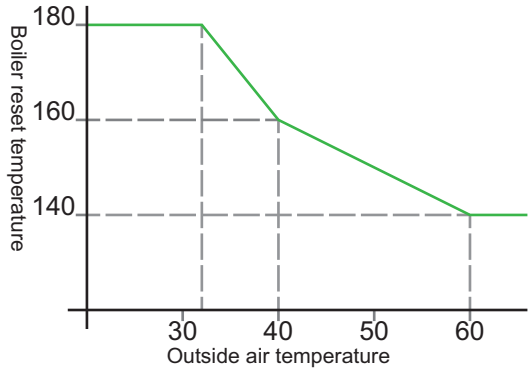


Table 29–16 Table values for boiler reset

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 330.

TIME

The TIME function that returns a value based on the time of day as maintained in the controller running Control Basic. The exact format for TIME varies between BACnet and KMD controllers.

> **Tip:** For the most accurate results, read the time only once in a program. If time is required in more than one place, read the time and then assign the value to a local variable.

BACnet time format In BACnet controllers, the function returns the value for system time as the number of seconds after midnight.

KMD time format In KMD controllers , the function returns a number in a 24-hour format.

BACnet example

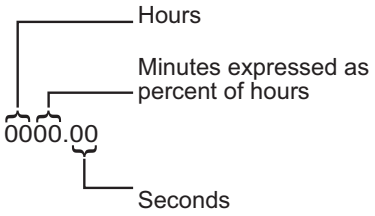
The following program returns hours, minutes, and seconds in local variables H, M, and S.

```
T = TIME
H = T \ 3600 : REM Hours
M = ( T - H * 3600 ) \ 60 : REM Minutes
S = T - 60 * M - 3600 * H : REM Seconds
```

KMD example

The TIME function in KMD controllers returns a number in the 24-hour format. For example 7:15:50 is returned by TIME as 725.30.

Illustration 29–4 TIME function format in KMD controllers



TIME can also be entered in the traditional hours:minutes:seconds format.

```
IF TIME = 12:30:00 THEN GOTO 20 ELSE END
REM it is after lunchtime
```

Time in the traditional hours:minutes:seconds format can be converted to a format to use in KMD controllers by using the following formula:

TIME = hour * 100 + (60 * min + sec) / 36

The following example calculates time from a KMD controller and breaks it down into hours, minutes, and seconds and stores the results in local variables.

```
10 T = TIME
20 H = T \ 100 : REM Hours
30 M = INT( ( T - H * 100 ) * 0.6 ) : REM Minutes
40 S = INT( 60 * ( T - INT( T ) ) ) : REM Seconds
```

TIMEOFF

Use the TIMEOFF function to determine if the present value of an object has been in the *Off* state for a specific period of time.

Syntax: *TIMEOFF(_point_)*

Note: For BACnet controllers only. For KMD controllers, see the keyword *TIME-OFF* on page 332.

Caution: TIMEOff responds to the time a point was *Off* as seen by the controller executing the program. This time may not be the same as the actual time if the point is from another controller.

The time used in the conditional can be expressed with any of the following formats:

hh:mm:ss	00:20:00
hh:mm	00:20
seconds	120

In the following example, binary output BO2 will be set to *Off* if output BO1 has been set to *Off* for more than 10 minutes.

```
IF TIMEOFF( BO1 )>0:10 THEN STOP BO2
```

See the related keyword topic *TIMEON* on page 333.

TIME-OFF

Use the TIME-OFF function to determine if a point has been in an *Off* state for a specific period of time.

Syntax: *TIME-OFF(_point_)*

- If *point* is a variable, Units must be configured as "Time".
- If *point* is a local variable, it will be returned as a number in the 24-hour format.

Note: Deprecated for BACnet controllers. See the topic *TIMEOFF* on page 332.

Caution: TIME-OFF responds to the time a point was *Off* as seen by the controller executing the program. This time may not be the same as the actual time if the point is from another controller.

The time used in the conditional can be expressed with either of the following formats:

hh:mm:ss	00:10:00
hh:mm	00:10

```
IF TIME-OFF( OUT1 ) > 0:10 THEN STOP OUT2
```

See the related topic *TIME-ON* on page 334 and *Programming with variables* on page 226.

TIMEON

Use the TIMEON function to determine if the present value of object has been in the *On* state for a specific period of time.

Syntax: *TIMEON(_point_)*



Note: For BACnet controllers only. For KMD controllers, see the keyword *TIME-ON* on page 334.



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.

The time used in the conditional can be expressed with any of the following formats:

hh:mm:ss	00:20:00
hh:mm	00:20
seconds	120

In the following example, binary 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 332.

TIME-ON

Use the TIME-ON function to determine if the present value in a point has been *On* for a specific period of time.

Syntax: *TIME-ON(_point_)*

- If *point* is a variable, Units must be configured as "Time".
- If *point* is a local variable, it will be returned as a number in the 24-hour format.

Note: Deprecated for BACnet controllers. See the keyword *TIMEON* on page 333.

Caution: TIME-ON responds to the time a point was *On* as seen by the controller executing the program. This time may not be the same as the actual time if the point is from another controller.

The time used in the conditional can be expressed with either of the following formats:

hh:mm:ss	00:10:00
hh:mm	00:10

```
IF TIME-ON( OUT1 ) > 0:10 THEN START OUT2
```

See the related topic *TIME-OFF* on page 332.

TOSTRING

Use TOSTRING to convert a numerical value to a string.

Syntax: *TOSTRING(argument1, argument2)*

See also the topic *STRING* on page 327.

TPAGE

Use the TPAGE command to initiate a telephone call to a text pager. *Telephone number* cannot contain spaces, hyphens, or other punctuation except commas. Each comma adds a two second pause.

Syntax: *TPAGE telephone number , text message*

Note: For KMDigital controllers only.

Note: KMC controllers do not support older TAPI protocols or speeds below 1200 baud.

```
DALARM VAR51 , 1 , VAR51 is in alarm!
ON-ALARM 40
GOTO 60
TPAGE 18005551212-5551212 , VAR 51 is in alarm! : REM
Line 40
WAIT 0:00:15 : ON-ERROR 40
END : REM Line 60
```

Note: Use TPAGE only in programs running in KMD Tier 1 controllers or stand-alone Tier 2 networks with a KMD-5559.

See the related topics *NPAGE* on page 301 and *PHONE* on page 309.

UNACK

Use UNACK to determine if there are alarms in the system which have not been acknowledged. The function returns *true* when an unacknowledged alarm exists or *false* if alarms have been acknowledged.

Note: For KMDigital controllers only.

```
IF UNACK THEN START VAR31 ELSE STOP VAR31
```

USER-A and USER-B

The USER-A and USER-B functions return the user number of any user logged on at local serial Port A. The functions return as *false* if an operator is not logged on.

Note: KMDigital Tier 1 controllers only.

```
IF+ USER-A > 0 THEN GOSUB 30
END
REM Line 30-Handles a user on Port A
RETURN
```

WAIT

Use WAIT to suspend execution in a Control Basic program for a specified time.

Syntax: *WAIT_period*

When Control Basic encounters a WAIT command, the following takes place:

- Program execution is suspended at the line with the WAIT command.
- A WAIT timer is started. The time is set to the value of *period* in the WAIT command.
- Control Basic moves to the next Control Basic program in the controller and no other lines in the program are evaluated.
- On the next scan, if the WAIT timer has not expired, Control Basic skips the entire program. The program with the WAIT command will continue to be skipped until the WAIT timer expires.
- When the WAIT timer expires, program execution will continue at the line following the WAIT statement.

Other programs in the controller will not be affected as WAIT applies only to the program in which it is listed.

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. For best results, place WAIT commands only at the end of a program.

Note: Do not use WAIT commands in programs that include INTERVAL, TIME-ON, or TIME-OFF statements. These statements require that the program runs on every scan to function properly.

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 330.

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
```

KMD example

```
START OUT2
REM Program continues here
WAIT 00:00:10 : REM ** Waits 10 seconds **
END
```

WRITE-TO-FLASH

Writes data to the non-volatile flash memory in the controller.

Note: For KMDigital controllers only.

Note: Firmware in the controller limits *WRITE-TO-FLASH* execution to once every 15 minutes. This limitation prevents *WRITE-TO-FLASH* commands from superseding other programming in the controller.

```
IF+ TIME > 17:00:00 THEN WRITE-TO-FLASH
IF ABS( VAR1 - VAR2 ) = 100 THEN WRITE-TO-FLASH
```

WS-OFF

Use the WS-OFF function to determine the time of day that a weekly schedule will change to *Off*. The parameter *schedule#* designates the weekly schedule and *time#slot* is the selected *Off* time, in earlier-to-later sequence.

Note: For KMDigital controllers only. See *SCHED-OFF* on page 318 for an equivalent Control Basic function in BACnet controllers.

Syntax: *WS-OFF(_schedule#_,_time#slot_)*

The following example sets variable VAR2 equal to the time that the schedule changes to *Off* for the third time on the day that Control Basic runs the statement.

```
VAR2 = WS-OFF( 1 , 3 )
```

WS-ON

Use the WS-ON function to determine the time of day that a weekly schedule will change to *On* . The parameter *schedule#* designates the weekly schedule and *time#slot* is the selected *On* time, in earlier-to-later sequence.



Note: For KMDigital controllers only. See *SCHEDON* on page 318 for an equivalent Control Basic function in BACnet controllers.

Syntax: *WS-ON(_schedule#_,_time#slot_)*

The following example sets variable VAR1 equal to the time that the schedule changes to *On* for the third time on the day that Control Basic runs the statement.

```
VAR1 = WS-ON( 1 , 3 )
```

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 225.

YEAR

Returns the four-place value of the current year.

```
Y = YEAR
```

Section 30: Writing block programs

This section is an overview of block programming in KMC Controls 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 Controls BACnet Controllers.

Topics in this section

- *Applicable controllers* on page 339
- *Starting the Block Editor* on page 341
- *A tour of the Block Editor* on page 343
- *Get and Set blocks* on page 344
- *Adding and editing routines* on page 346
- *Defining and using local variables* on page 345
- *Macro blocks* on page 347
- *Limitations and importing line programs* on page 350

Applicable controllers

Block programming is available only in the KMC Conquest series of controllers. This includes the models from the following list.

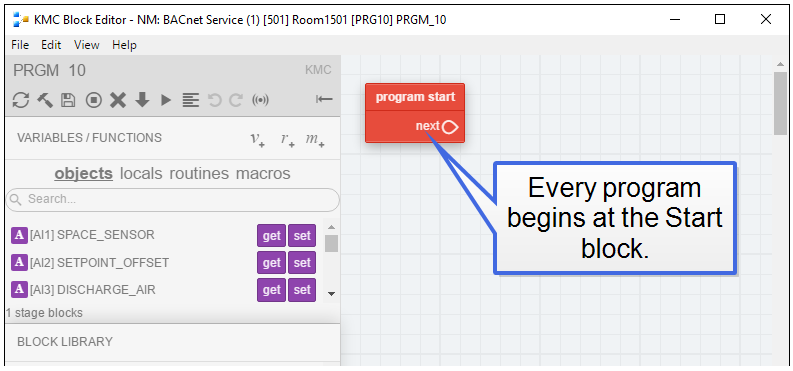
- BAC-5901
- BAC-9001
- BAC-9021
- BAC-9301
- BAC-9311

Starting the Block Editor

To start block programing do one of the following:

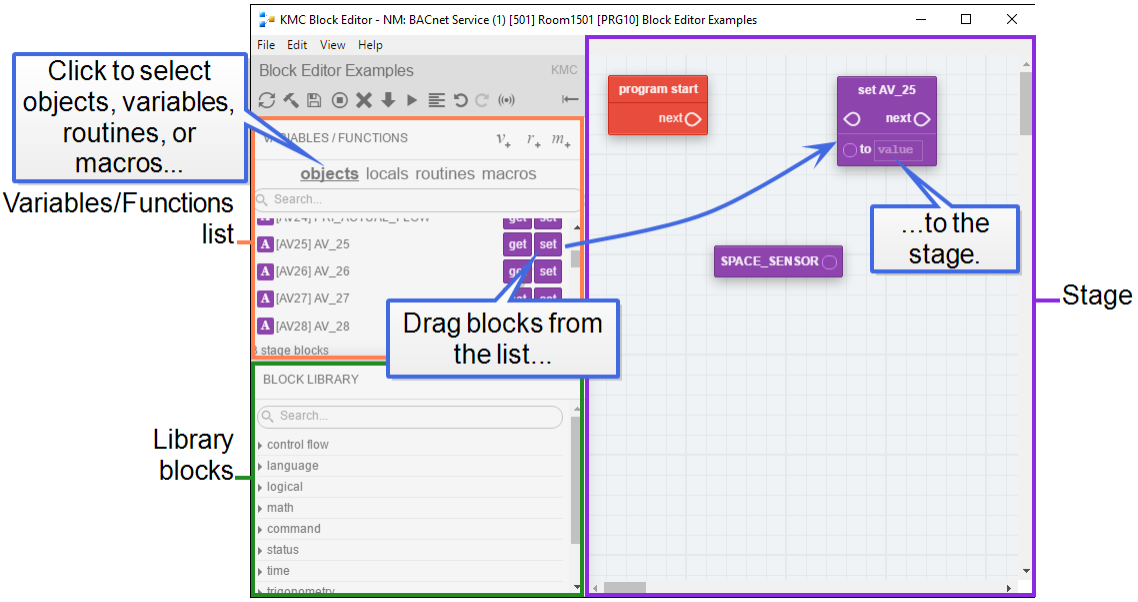
- In the Network Manager list, right-click a program object icon and then choose **Block Editor** from the shortcut menu.
- Open a program object and then choose **Block Editor** from the Control Basic Program group.

The Block Editor always includes a red Program Start block. The Start block is added automatically and cannot be deleted.

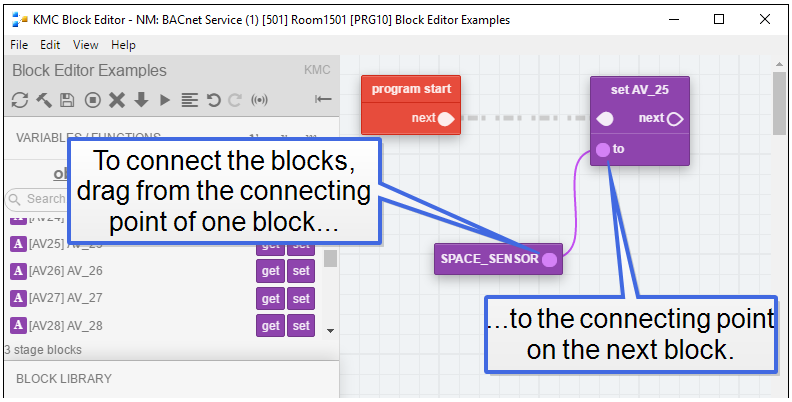


Add program blocks to the stage by dragging them from the variables and objects list or from one of the block library groups.

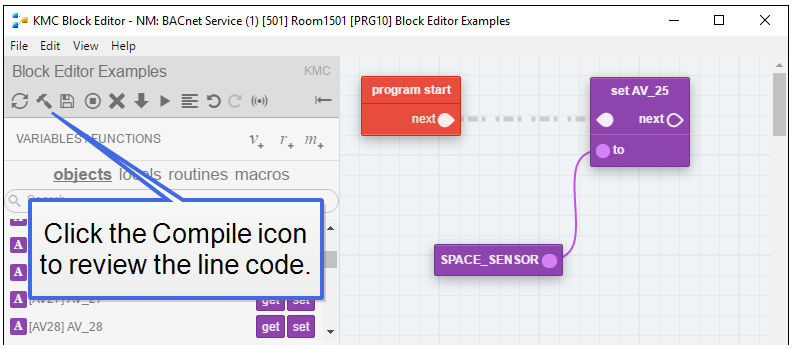
- Blocks in the library are programming functions and commands.
- The Get and Set blocks in the Variables/Functions list transfer values to a program step.



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 `AV14="SPACE_SENSOR"`.

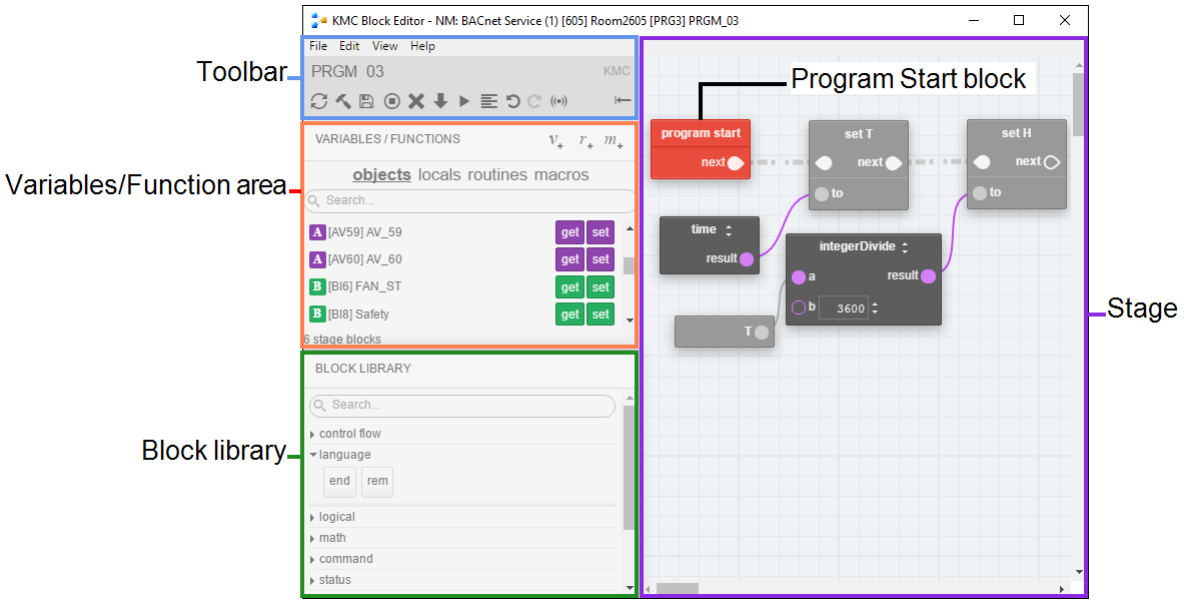


Once all blocks are in place and connected, click the Compile icon on the tool bar.



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. See the topic *The Block Editor toolbar* on page 344 for other toolbar commands.

Variables/Function area This area has three parts to select blocks or add routines, variables, and macros.

- Use the three buttons at the top part of the area to add local variables, routines, or macros.
- Select *objects*, *locals*, *routines*, or *macros* to change the type of blocks displayed in the list.
- The blocks in the list represent objects in the controllers and local variables. If defined, subroutine and macro blocks may also be in the Variables/Function area.

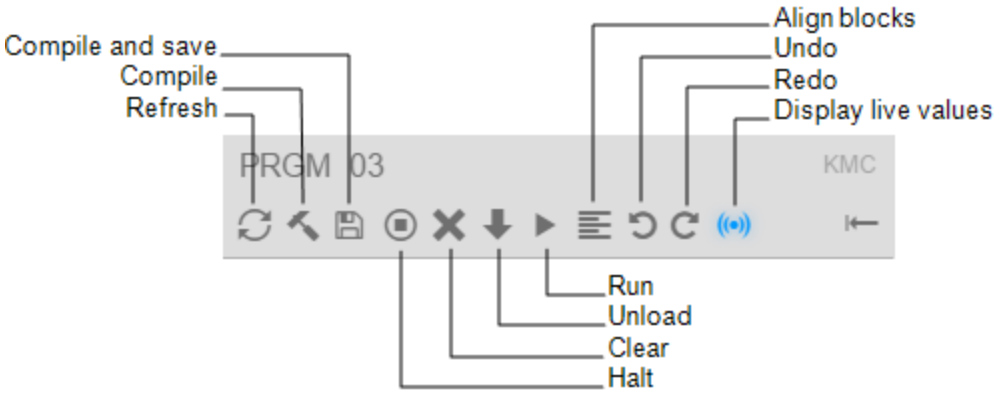
Stage The stage is the work area of the Block Editor. Blocks are added to the stage to build the program.

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 353.

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 Controls controllers this is similar to Clear.

Run Compiles the program, saves it to memory, and runs the program in the program object.

Align blocks Select two or more blocks to align the tops, left sides, right sides or bottoms of the selected blocks.

Undo Undo the last action.

Redo Redo the last action.

Display live values Displays the actual value of the block's selected property. If the property type changes, compile the program to display the value of the new property.

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.

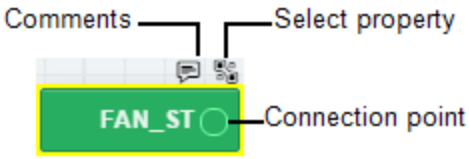
Get blocks

A Get block transfers a value to a program statement from local variables or objects in the controller.

Comments Save notes or remarks for the 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 9.

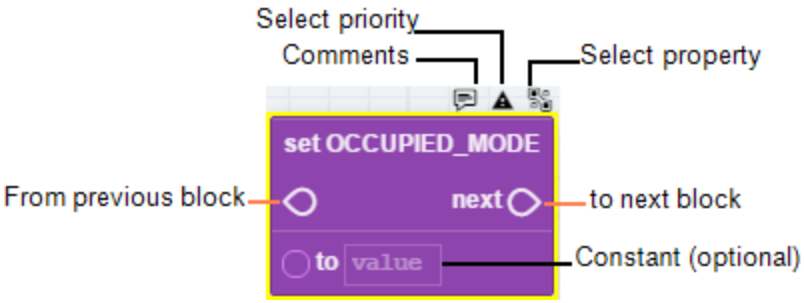
Connection point Connect this point to Set blocks or the input on other types of blocks.

From previous block Connects 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 Connects to a Get block or the *result* from another block such as a Math block. The *to* connector is disabled when there is an entry in *value*.



Defining and using local variables

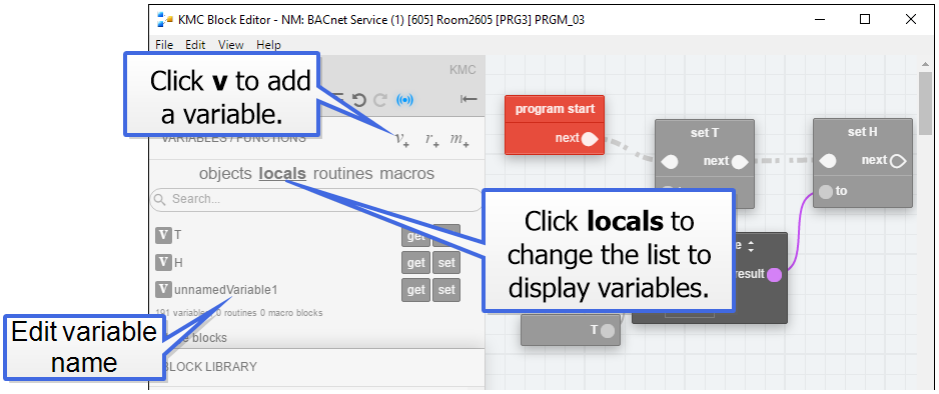
Local variables are place holders for information. They can only be used within the Program object that defines them and cannot be directly transferred to other Control Basic programs. Local variables are often used for counters or to store the results of local calculations.

Adding new variables To add a new local variable, do the following:

1. Click the variable button in the Variables/Function area. The Variables/Function list automatically changes to Variables. New 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.

Using an existing variable To use the Get or Set block of an existing local variable, do the following:

1. Click **locals** to change the Variables/Function list to display local variables.
2. Drag the Get or Set blocks to the Stage.

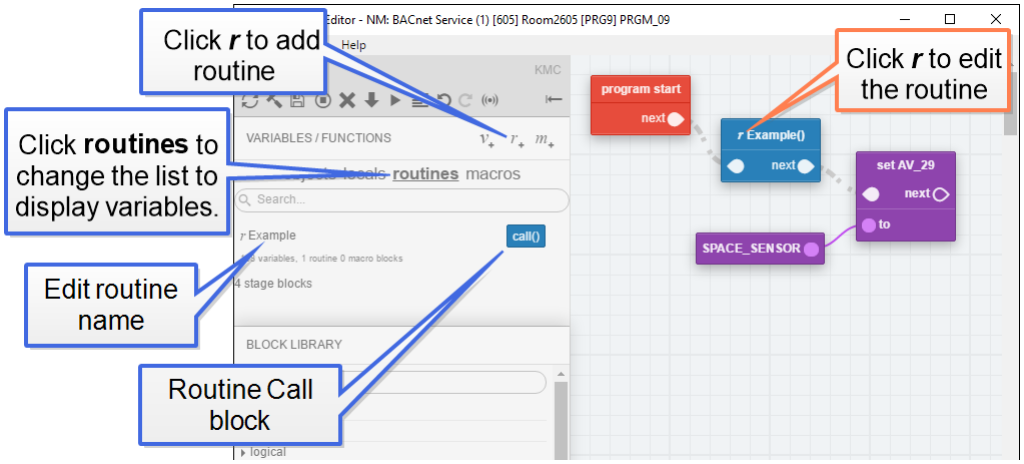


Adding and editing routines

A routine—often referred to as a subroutine— is a packaged unit of program blocks that performs a specific task that is repeated within the same program. The unit can then be added to a program wherever that particular task is required. Routines are added as Call blocks in the Block Editor.

To add a routine, do the following:

1. Click the routine button r+ in the Variables/Function area. A Call block is added to the routines list and the list automatically changes to the routines 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.

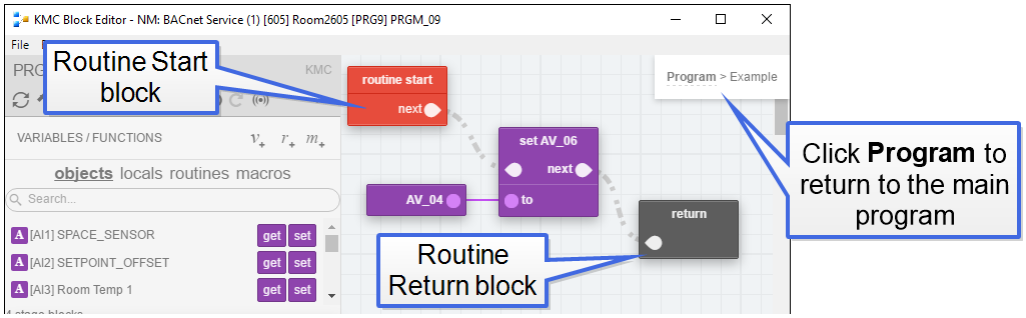


To add a Call block to a program, do the following:

1. Click **routines** to change the Variables/Function list to display the list of Call blocks.
2. Drag a Call block to the Stage.

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 or other programs without rewriting complicated or long sequences.

Tip: A properly written Macro block can be moved to another program without rewriting it because it does not directly refer to controller objects. All references to objects are made through the Macro block ports.

Macro block ports are the inputs and outputs to the set of instructions. The ports represent the arguments or parameters used within the macro.

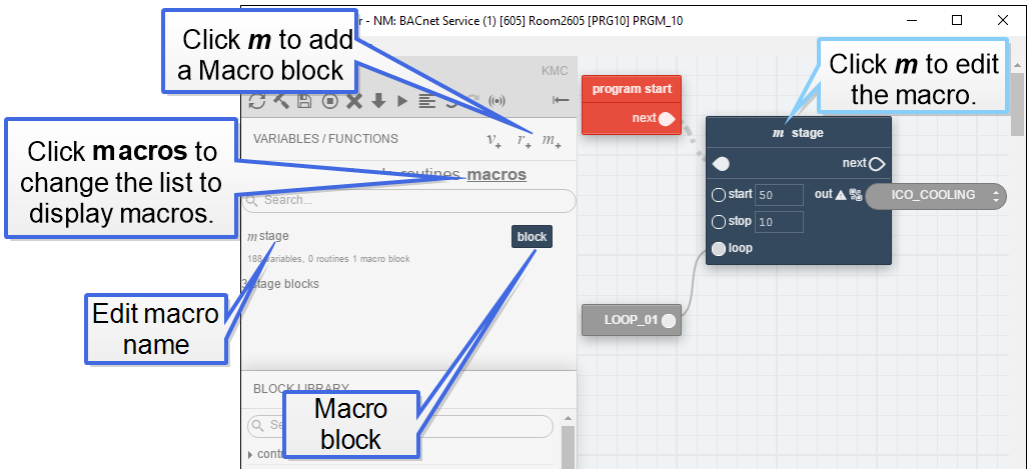
- Input ports transfer values into the macro.
- Output ports transfer results from the macro to variables.
- The ports appear as the connector points on the Macro block.
- Ports are optional and can be added, deleted, or renamed as needed when setting up the macro.
- When the Macro block is compiled, the reference to the port is replaced by the actual parameter, such as the results from a previous block, local variable, or constant.

Macro blocks can be saved on the local computer and reused in other programs, either in the same controller or in other controllers in the Network Manager list.

Adding a new Macro block

To add a new Macro block to the program, do the following:

1. Click the Macro button m_+ in the Variables/Function area. A Macro block is added to the Macros list and the list automatically changes to show the Macro blocks.
2. Give the new 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.



Adding existing Macro blocks to the stage

To add an existing Macro block to the stage, do the following:

1. Click **macros** to change the Variables/Function list to display the Macro blocks.
2. Drag the Macro block to the stage.
3. Connect the Macro block to other blocks as required by program logic.

Editing and saving Macro blocks

Macro blocks can only be edited on the stage. To edit an existing Macro block, do the following:

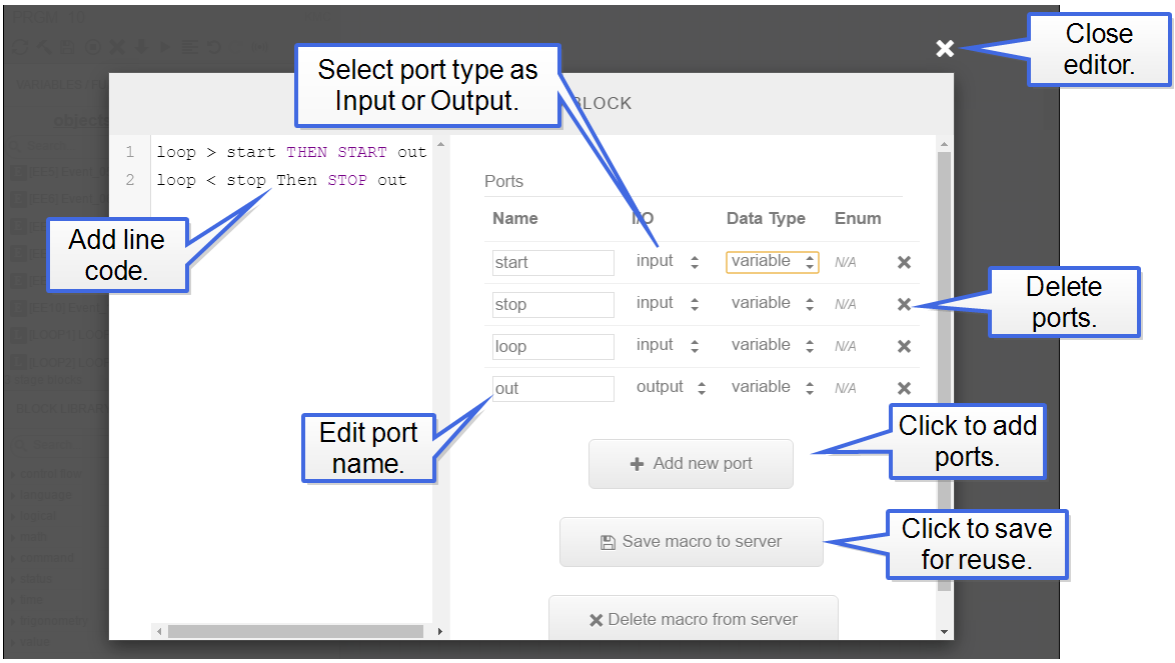
1. On a Macro block, click the **m** next to the macro name.
2. In the Macro Block Editor, add or delete ports as needed.
   - Input ports add input connectors to the macro block and transfer parameters into the line code.
   - Output ports add a variable selector to the macro block and transfer results 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 level 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 block for reuse in other programs, click **Save macro to server**.

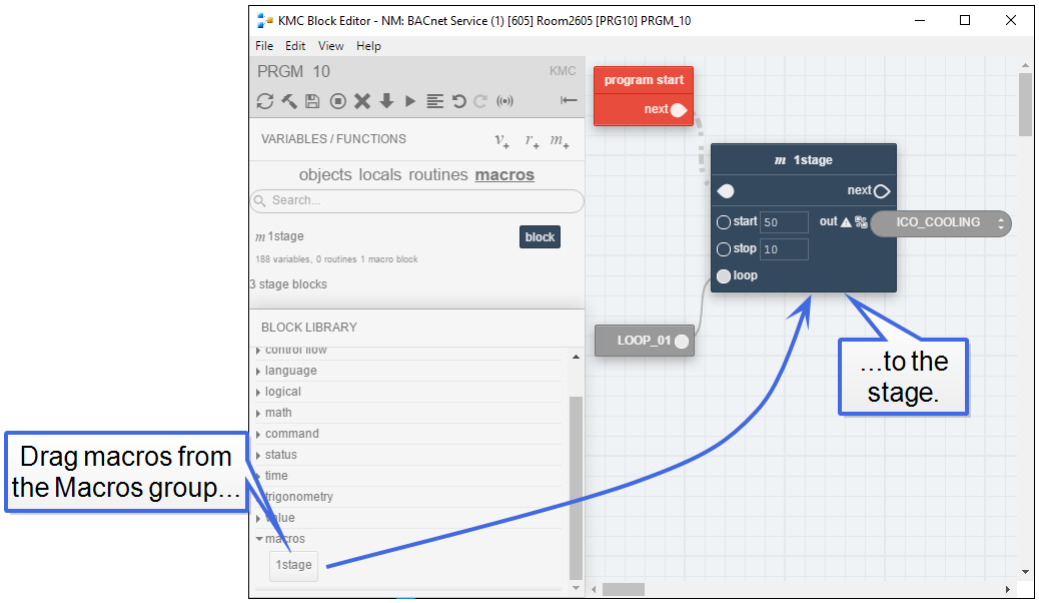
5 When finished, click the close editor button ⊠.



Reusing Macro blocks

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 Block 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 line programs

The Block editor will attempt to import programs from existing line code. However, some programs may be too complex and will require revision before they can be imported.

Note: Because of the limitations of importing line programs, switching between line programing and block programming in the same program object is not recommended.

Capitalization All keywords in the line program must be capitalized before importing.

REM Remark statements are not imported. They can be added in the Block Editor once the program is imported.

Multiple statements on one line Statements on the same line that are 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. See the topics *Transferring values between BACnet controllers* on page 232.

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

Section 31: Reference to Control Basic blocks

This section covers the blocks for the Control Basic block programming.

Abs block

Block type: Math

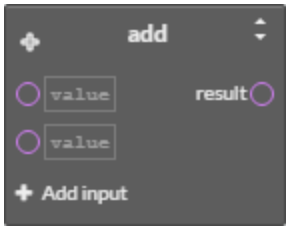
The Abs block returns the absolute value of the input. The input at *a* can be a Get block or the output of a Math block. Connect the *result* connector to another Math block or a Set block.



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.

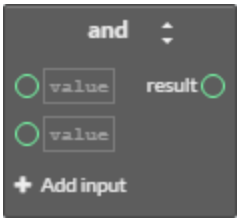
- *Multiply block* on page 380
- *Divide block* on page 366
- *IntegerDivide block* on page 374
- *Mod block* on page 379
- *Subtract block* on page 393
- *Power block* on page 383

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 382 and *Xor block* on page 397.

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.



If the time value at *MinOffTime* is a constant, it can be expressed with any of the following formats:

hh:mm:ss	00:20:00
hh:mm	00:20
seconds	120

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.



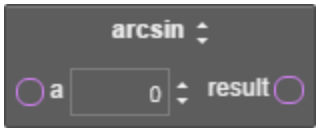
The Arccos block can be changed to any of the following types of blocks by clicking on the double-arrows and choosing a new function.

- *Arcsin block* on page 355
- *Arctan block* on page 356
- *Cos block* on page 361
- *Sin block* on page 391
- *Tan block* on page 394

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.

- *Arccos block* on page 355
- *Arctan block* on page 356
- *Cos block* on page 361
- *Sin block* on page 391
- *Tan block* on page 394

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.

- *Arccos block* on page 355
- *Arcsin block* on page 355
- *Cos block* on page 361
- *Sin block* on page 391
- *Tan block* on page 394

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 378.

Clear block

Block type: Command

The Clear block resets the value of all local variables to zero.



Close block

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 block* on page 382
- *Start block* on page 392
- *Stop block* on page 393
- *Enable block* on page 368
- *Disable block* on page 365

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*.



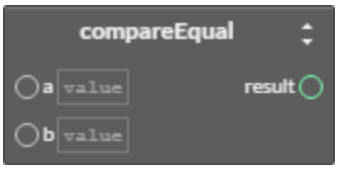
See the following topics for additional comparison blocks.

- *CompareEqual block* on page 358
- *CompareGreater block* on page 358
- *CompareGreaterEqual block* on page 359
- *CompareLess block* on page 359
- *CompareLessEqual block* on page 360
- *CompareNotEqual block* on page 361

CompareEqual block

Block type: Logical

The CompareEqual block tests whether one value is 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 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 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 block* on page 358
- *CompareGreaterEqual block* on page 359
- *CompareLess block* on page 359
- *CompareLessEqual block* on page 360
- *CompareNotEqual block* on page 361

See also the topic *Compare block* on page 357.

CompareGreater block

Block type: Logical

The CompareGreater block tests if one value is greater than 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 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 CompareGreater block can be changed to any of the following types of blocks by clicking on the double-arrows and choosing a new function.

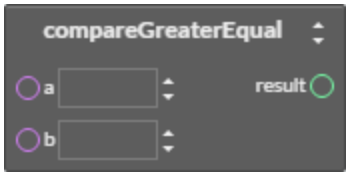
- *CompareEqual block* on page 358
- *CompareGreaterEqual block* on page 359
- *CompareLess block* on page 359
- *CompareLessEqual block* on page 360
- *CompareNotEqual block* on page 361

See also the topic *Compare block* on page 357.

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. The block is typically used as the *condition* for an *If block* or connected to a Set block.



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 CompareGreaterEqual block can be changed to any of the following types of blocks by clicking on the double-arrows and choosing a new function.

- *CompareEqual block* on page 358
- *CompareGreater block* on page 358
- *CompareLess block* on page 359
- *CompareLessEqual block* on page 360
- *CompareNotEqual block* on page 361

See also the topic *Compare block* on page 357.

CompareLess block

Block type: Logical

The CompareLess block tests if one value is less than 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 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 CompareLess block can be changed to any of the following types of blocks by clicking on the double-arrows and choosing a new function.

- *CompareLess block* on page 359
- *CompareGreater block* on page 358
- *CompareGreaterEqual block* on page 359
- *CompareLessEqual block* on page 360
- *CompareNotEqual block* on page 361

See also the topic *Compare block* on page 357.

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 block* on page 358
- *CompareGreater block* on page 358
- *CompareGreaterEqual block* on page 359
- *CompareLess block* on page 359
- *CompareNotEqual block* on page 361

See also the topic *Compare block* on page 357.

CompareNotEqual block

Block type: Logical

The CompareNotEqual blocks tests whether one value is not equal to a second value. 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. If the values at *a* and *b* are not equal, the result is logically true.

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 block* on page 358
- *CompareGreater block* on page 358
- *CompareGreaterEqual block* on page 359
- *CompareLess block* on page 359
- *CompareLessEqual block* on page 360

See also the topic *Compare block* on page 357.

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.

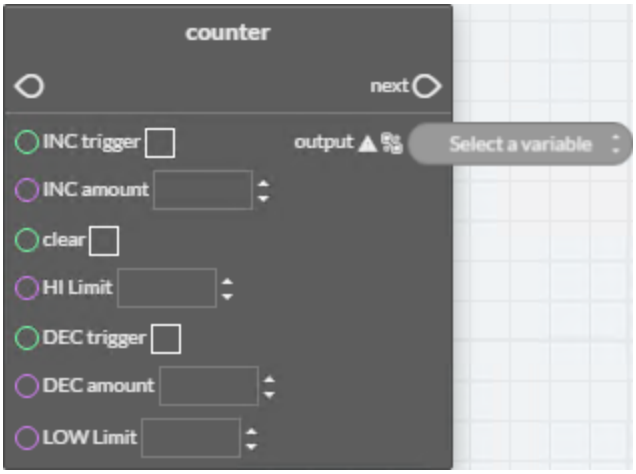
- *Arccos block* on page 355
- *Arcsin block* on page 355
- *Arctan block* on page 356
- *Sin block* on page 391
- *Tan block* on page 394

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.



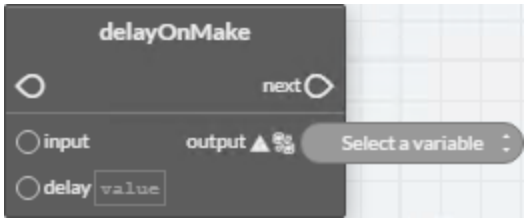
DelayOnMake block

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.

If the time value at *delay* is a constant, it can be expressed with any of the following formats:

hh:mm:ss	00:20:00
hh:mm	00:20
seconds	120

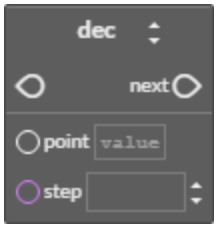


See also the topic *DelayOnBreak block* on page 363.

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.

If the time value at *delay* is a constant, it can be expressed with any of the following formats:

hh:mm:ss	00:20:00
hh:mm	00:20
seconds	120



See also the topic *DelayOnMake block* on page 362.

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 block* on page 368
- *Enthalpysi block* on page 369
- *Dewpointsi block* on page 365

To calculate dew point in degrees Celsius, see the topic *Dewpointsi block* on page 365.

Dewpointsi block

Block type: Value

The Dewpointsi block returns the dew point in degrees Celsius based on outside air temperature, in degrees Celsius, and the outside 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 Dewpointsi block can be changed to any of the following types of blocks by clicking on the double-arrows and choosing a new function.

- *Dewpoint block* on page 364
- *Enthalpy block* on page 368
- *Enthalpysi block* on page 369

To calculate dew point in degrees Fahrenheit, see the topic *Dewpoint block* on page 364.

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.



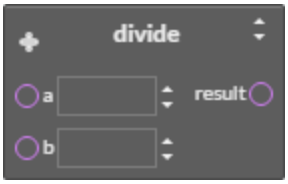
The Disable block can be changed to any of the following types of blocks by clicking on the double-arrows and choosing a new command.

- *Close block* on page 357
- *Open block* on page 382
- *Start block* on page 392
- *Stop block* on page 393
- *Enable block* on page 368

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 block* on page 380
- *IntegerDivide block* on page 374
- *Mod block* on page 379
- *Add block* on page 353
- *Subtract block* on page 393
- *Power block* on page 383

See the related topics *IntegerDivide block* on page 374 and *Mod block* on page 379.

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 block* on page 367
- *Doy block* on page 367
- *Month block* on page 379
- *Time block* on page 395
- *Year block* on page 398

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 block* on page 366
- *Doy block* on page 367
- *Month block* on page 379
- *Time block* on page 395
- *Year block* on page 398

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. This 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 block* on page 366
- *Dow block* on page 367
- *Month block* on page 379
- *Time block* on page 395
- *Year block* on page 398

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 block* on page 357
- *Open block* on page 382
- *Start block* on page 392
- *Stop block* on page 393
- *Disable block* on page 365

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 may 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 of *result* is expressed as BTUs per pound of air. 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 Enthalpy block can be changed to any of the following types of blocks by clicking on the double-arrows and choosing a new function.

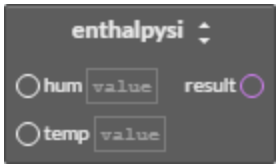
- *Enthalpysi block* on page 369
- *Dewpoint block* on page 364
- *Dewpointsi block* on page 365

To calculate enthalpy as kilojoules per kilogram of air, see the topic *Enthalpysi block* on page 369.

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 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-arrow and choosing a new function.

- *Enthalpy block* on page 368
- *Dewpoint block* on page 364
- *Dewpointsi block* on page 365

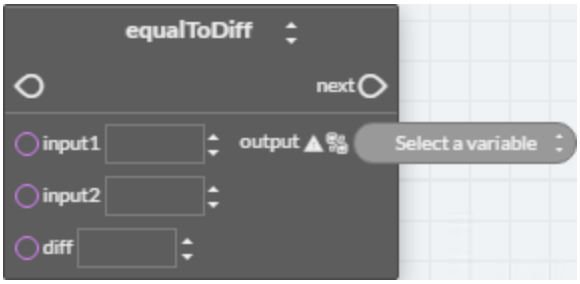
To calculate enthalpy as BTUs per pound of air, see the topic *Enthalpy block* on page 368.

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*, the output variable is *FALSE*.
- 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 required.
- Only binary objects or local variables can be selected for *output*.

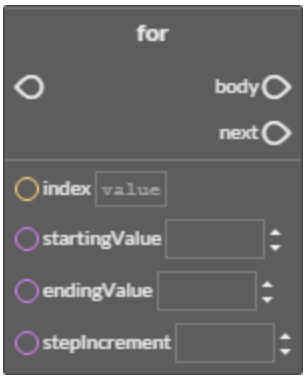


The EqualToDiff block can be changed to a *LessThanDiff block* on page 376or a *GreaterThanDiff block* on page 371 by clicking on the double-arrows and choosing a new function.

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 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*.

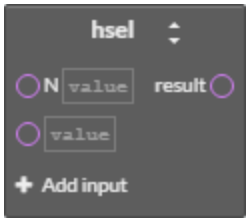


The GreaterThanDiff block can be changed to an *EqualToDiff block* on page 370 or a *LessThanDiff block* on page 376 by clicking on the double-arrows and choosing a new function.

Hsel block

Block type: Logical

The Hsel block selects the highest, second highest, etc. value of blocks connected to the input. 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**.

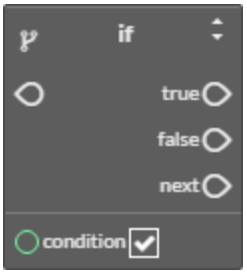
See also the topic *Lsel block* on page 377.

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 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 372 and *IfPlus block* on page 373.

IfMinus block

Block type: Control flow

The IfMinus 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.

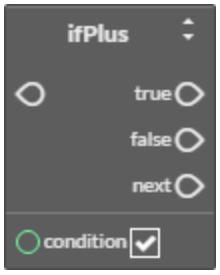
See also the topics *If block* on page 372 and *IfPlus block* on page 373.

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 372 and *IfMinus block* on page 372.

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.

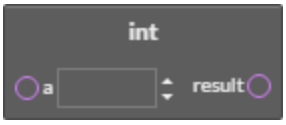


The Inc block can be changed to a *Dec block* by clicking on the double-arrows and choosing the new function.

Int block

Block type: Math

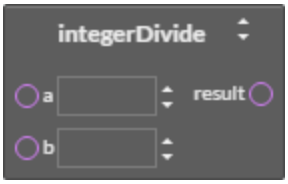
The Int block returns the integer portion of the block connected to *a*. The value returned is the greatest integer that is less than or equal to the value of *expression*. The value at *a* can be a constant, a Get block, or the result of a Math block. Connect the *result* connector to another Math block or a Set block.



IntegerDivide block

Block type: Math

The IntergerDivide block divides the value of *a* by the value of *b* and returns 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 block* on page 380
- *Divide block* on page 366
- *Add block* on page 353
- *Mod block* on page 379
- *Subtract block* on page 393
- *Power block* on page 383

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. If the time value at *a* is a constant, it can be expressed with any of the following formats:

hh:mm:ss	00:20:00
hh:mm	00:20
seconds	120



Invln block

Block type: Math

The Invln 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*.



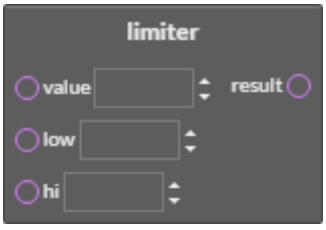
The LessThanDiff block can be changed to a *GreaterThanDiff block* on page 371 or an *EqualToDiff block* on page 370 by clicking on the double-arrows and choosing a new function.

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.

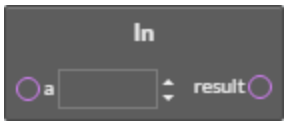
Tip: Connecting a Get block to *value* and a Set block to *result* from the same object tests and limits the object's value.



Ln block

Block type: Math

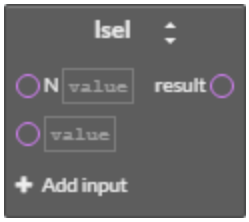
The Ln block returns the natural logarithm of the value at *a*. The value at *a* can be a constant, a Get block, or the result of a Math block. Connect the *result* connector to another Math block or a Set block.



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. 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 371.

Max block

Block type: Math

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 block* on page 356 or *Min block* on page 378 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: Math

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 block* on page 356 Avg or *Max block* on page 377 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: Math

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 356
- *Max block* on page 377
- *Min block* on page 378

Mod block

Block type: Math

The Mod block returns the remainder (or modulus) of *a* divided by *b*. The values at *a* and *b* can be from constants, Get blocks, or the result from a Math block. 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.

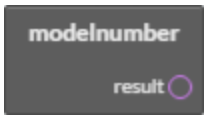
- *Multiply block* on page 380
- *Divide block* on page 366
- *IntegerDivide block* on page 374
- *Add block* on page 353
- *Subtract block* on page 393
- *Power block* on page 383

See the related topics *Divide block* on page 366 and *IntegerDivide block* on page 374.

Modelnumber block

Block type: Status

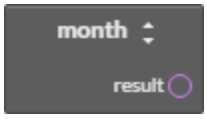
Use the Modelnumber block to set a variable or property to the numerical portion of the model number of the controller. The block is typically used with an IF block, a Logical block, or connected to a Set block.



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.

- *Dow block* on page 367
- *Doy block* on page 367
- *Doy block* on page 367
- *Time block* on page 395
- *Year block* on page 398

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 block* on page 366
- *IntegerDivide block* on page 374
- *Mod block* on page 379
- *Add block* on page 353
- *Subtract block* on page 393
- *Power block* on page 383

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

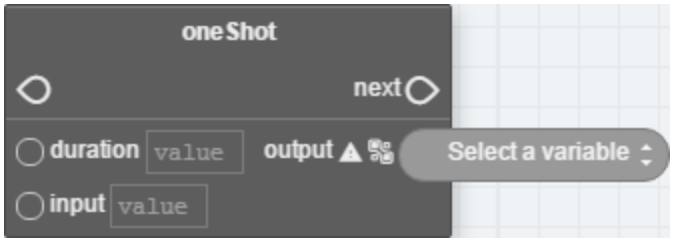
The Not block performs a logical negation operation on the input. If the input is 0, the result is 1. If the input is non-zero, the result is 0. The block at *a* can be from a Get block or the result of a Math block.



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.



If the time value at *duration* is a constant, it can be expressed with any of the following formats:

hh:mm:ss	00:20:00
hh:mm	00:20
seconds	120

See also the topic *IfPlus block* on page 373.

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 block* on page 357
- *Start block* on page 392
- *Stop block* on page 393
- *Enable block* on page 368
- *Disable block* on page 365

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 block* on page 354 or *Xor block* on page 397 by clicking on the double-arrows and choosing the new function.

Add additional inputs to the block by clicking **+ Add Input**.

Outputoverride block

Block type: Status

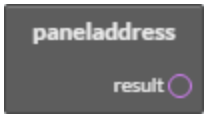
The Outputoverride block returns the switch position of an optional HPO-6700 series output board installed in the controller in which Control Basic is running. The block returns *FALSE* if the switch is in *AUTO* and *TRUE* if the switch is set to either the *OFF* or *HAND* position. The output object 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.



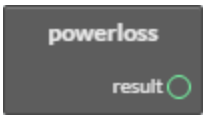
The Power block can be changed to any of the following types of blocks by clicking on the double-arrows and choosing a new function.

- *Multiply block* on page 380
- *Divide block* on page 366
- *IntegerDivide block* on page 374
- *Mod block* on page 379
- *Add block* on page 353
- *Subtract block* on page 393

Powerloss block

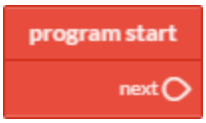
Block type: Status

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 marks 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

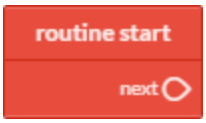
Block type: Math

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

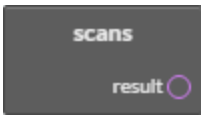
The Routine Start block is automatically added to new routines. The block is the beginning of the routine and cannot be removed. See the topic *Adding and editing routines* on page 346.



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 the program longer to process and the scan rate decreases. The value at *result* is expressed in scans per second.



Schedoff block

Block type: Time

Use the Schedoff block to find when a schedule object will next set its reference object to *INACTIVE* or a value of zero (0). The schedule object must be within the same controller as the controller running the program. The value returned at *result* is for the current day, as maintained in the controller, of the weekly schedule in a schedule object.



Rules for the Schedoff block:

- The connection at *schedule* can be a local variable Get block, an analog value from an object Get block, or a fixed value.
- The connection at *time* may be specified with a local variable Get block, an analog value from an object Get block, a fixed value, or a Time block. If a Time block is used, the time maintained in the controller running the program is the time value used.
- To use a fixed value at *time*, enter it in the 24-hour format (hh:mm:ss).
- The value at *result* is the difference—in seconds—between the value specified at *time* and the next scheduled time that the schedule's present value will change to *INACTIVE* (0).
- A value of 0 at *result* indicates that the schedule is already set to *INACTIVE*.
- A value of 86,400 at *result* indicates that there are no more scheduled *INACTIVE* times for the current day.

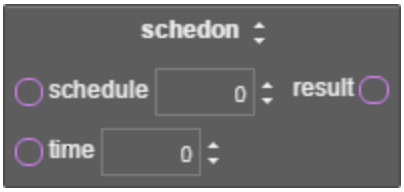
The Schedoff block can be changed to a Schedon block by clicking on the double-arrows and choosing the new function.

See also the topic *Schedon block* on page 388.

Schedon block

Block type: Time

Use the Schedon block to find when a schedule object will next set its reference object to *ACTIVE* or a non-zero value. The schedule object must be within the same controller as the controller running the program. The value returned at *result* is for the current day, as maintained in the controller, of the weekly schedule in a schedule object.



- The connection at *schedule* can be a local variable Get block, an analog value from an object Get block, or a fixed value.
- The connection at *time* may be specified with a local variable Get block, an analog value from an object Get block, a fixed value, or a Time block. If a Time block is used, the time maintained in the controller running the program is the time value used.
- To use a fixed value at *time*, enter it in the 24-hour format (hh:mm:ss).
- The value at *result* is the difference—in seconds—between the value specified at *time* and the next scheduled time that the schedule's present value will change to *ACTIVE* or a non-zero(0) value.
- A value of 0 at *result* indicates that the schedule is already set to *ACTIVE* or a non-zero(0) value.
- A value of 86,400 at *result* indicates that there are no more scheduled *ACTIVE* or a non-zero(0) values scheduled for the current day.

The Schedon block can be changed to a Schedoff block by clicking on the double-arrows and choosing the new function.

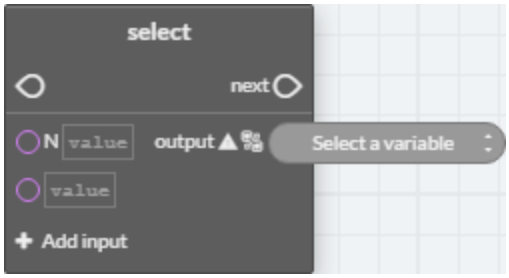
See also the topic *Schedoff block* on page 387.

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 constants.
- The variable for *output* must be an analog object or a local variable.

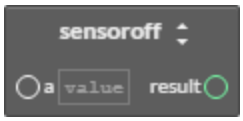


Add additional inputs to the block by clicking **+ Add Input**.

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 390 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 389 and *If block* on page 372, *IfMinus block* on page 372, or *IfPlus block* on page 373, 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).

A Sensoroff block 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 31–1 Example table for Sensoroff blocks

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 blocks* on page 390, 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 the block 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 that 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 389 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 389and *If block* on page 372, *IfPlus block* on page 373, or *IfMinus block* on page 372, 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 31–2 Input conditions for Sensoron blocks

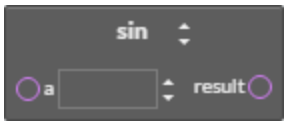
Input Voltage	Detected condition
0	Closed circuit
0.4	Temperature-55 degrees
4.9	Temperature-95 degrees
5.0 or greater	Open circuit

In the table *Input conditions for Sensoron blocks*, 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 (0), which is a condition the block 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 that 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 block* on page 355
- *Arcsin block* on page 355
- *Arctan block* on page 356
- *Cos block* on page 361
- *Tan block* on page 394

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 block* on page 357
- *Open block* on page 382
- *Stop block* on page 393
- *Enable block* on page 368
- *Disable block* on page 365

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.



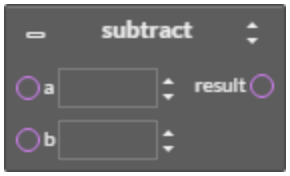
The Stop block can be changed to any of the following types of blocks by clicking on the double-arrows and choosing a new command.

- *Close block* on page 357
- *Open block* on page 382
- *Start block* on page 392
- *Disable block* on page 365
- *Enable block* on page 368

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.



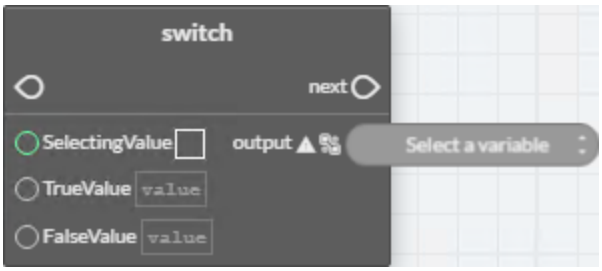
The Subtract block can be changed to any of the following types of blocks by clicking on the double-arrows and choosing a new function.

- *Multiply block* on page 380
- *Divide block* on page 366
- *IntegerDivide block* on page 374
- *Mod block* on page 379
- *Add block* on page 353
- *Power block* on page 383

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 block* on page 355
- *Arcsin block* on page 355
- *Arctan block* on page 356
- *Cos block* on page 361
- *Sin block* on page 391

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.

- *Dow block* on page 367
- *Doy block* on page 367
- *Dom block* on page 366
- *Month block* on page 379
- *Year block* on page 398

Timeoff block

Block type: Time

Use the Timeoff block to determine if the present value of the object at *point* has been in the *Off* state for a specific period of time. The value returned at *result* is connected to one port of a Compare block and then compared to a time value.



The Timeoff block can be changed to a Timeon block by clicking on the double-arrows and choosing the new function.

See also the topic *Timeon block* on page 396.

Timeon block

Block type: Time

Use the Timeon block to determine if the present value of an object has been in the *On* state for a specific period of time. The value returned at *result* is connected to one port of a Compare block and then compared to a time value.

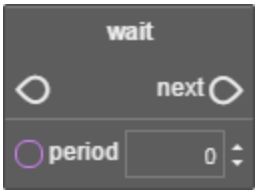


The Timeon block can be changed to a Timeoff block by clicking on the double-arrows and choosing the new function.

See also the topic *Timeoff block* on page 395.

Wait block

Block type: Time



Use a WAIT block to suspend execution in a Control Basic program for a specified time. The time at *period* can be a Get block, the output of a Math block, or a constant. If the time value at *period* is a constant, it can be expressed with any of the following formats:

hh:mm:ss	00:20:00
hh:mm	00:20
seconds	120

When Control Basic encounters a WAIT block, the following takes place:

- Program execution is suspended at the WAIT block.
- A WAIT timer is started. The time is set to the value of *period* in the WAIT block.
- Control Basic moves to the next Control Basic program in the controller and no other blocks in the program are evaluated.
- On the next scan, if the WAIT timer has not expired, Control Basic skips the entire program. The program with the WAIT block will continue to be skipped until the WAIT timer expires.
- When the WAIT timer expires, program execution will continue at the next block.

Other programs in the controller will not be affected as WAIT applies only to the program in which it is listed.

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. For best results, place WAIT statements only at the end of a program.

Note: Do not use WAIT statements in programs that include INTERVAL, TIME-ON, or TIME-OFF blocks. These blocks require that the program runs on every scan to function properly.

Within block

Block type: Logical

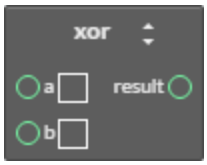
The Within block is a macro block that sets the output to *ACTIVE (1)* if the input at *value* is within the range set by *HI* and *LO*. If value is not within the range, the output is set to *INACTIVE (0)*. The inputs at *variable*, *HI*, and *LO* can be from constants, a Get block, or the results from 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.

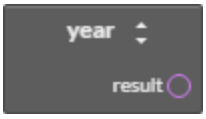


The Xor block can be changed to an *And block* on page 354 or an *Or block* on page 382 by clicking on the double-arrows and choosing the new function.

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.

- *Dow block* on page 367
- *Doy block* on page 367
- *Dom block* on page 366
- *Month block* on page 379
- *Time block* on page 395

KMC Connect

Part VII: BACnet objects

Section 32: 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 ASHRAE 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 401
- *BACnet services* on page 402
- *Priority arrays* on page 402
- *Configuring BACnet controllers for alarms* on page 135
- *Reference to BACnet objects* on page 435

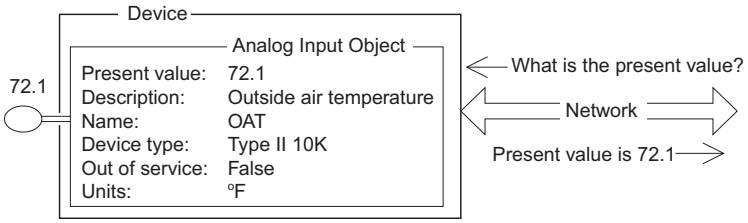
BACnet objects and properties

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 401.

Illustration 32–1 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 401 , 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

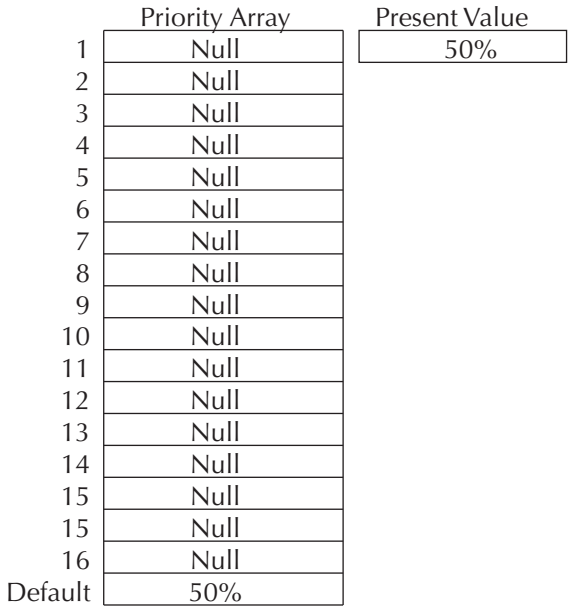
BACnet devices use the priority array to control the Present Value property in output and value objects. The priority array property maintains order when several commands are simultaneously issued to change the Present Value property. For example, an operator may enter a command to stop a fan when a 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

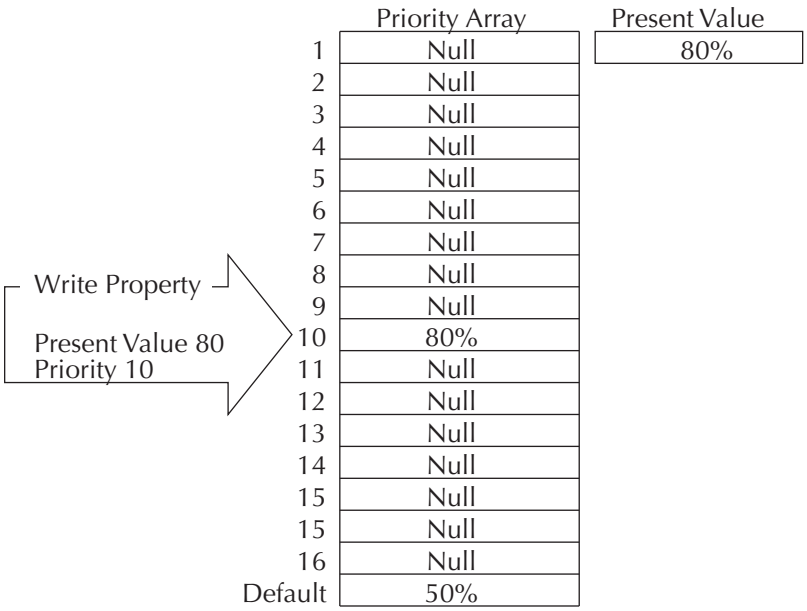
Initially, all levels of a priority array are filled with a *NULL* value. The default value is entered in Relinquish Default in the object menu. In this example, Relinquish Default equals 50%.

Illustration 32–2 Initial state



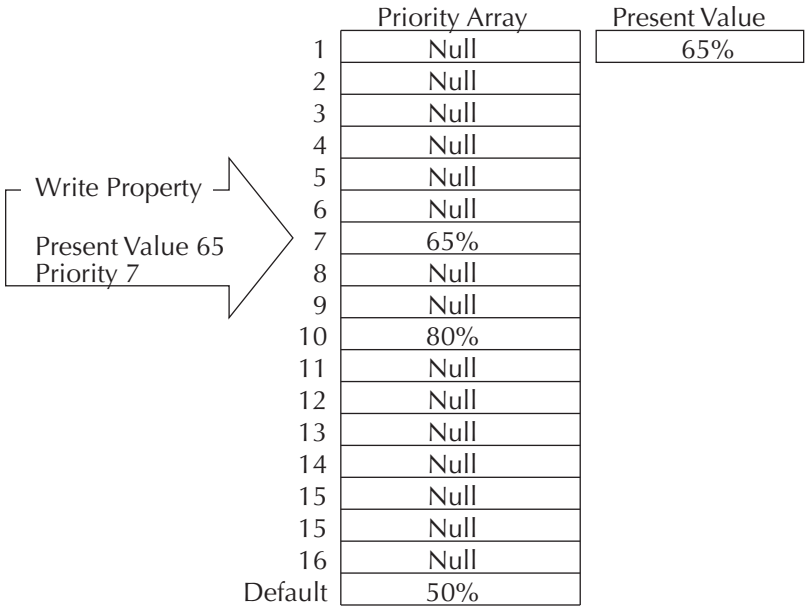
Then, a write property command with a value of 80% at a priority level of 10 is sent to the object. Because this new priority is a higher priority than the default level, the present value changes to 80%. Note that the array default value remains in the array.

Illustration 32–3 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 32–4 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 315.

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. By standardizing the priority levels, the levels are applied consistently across various vendors, types of facilities, and applications.

Table 32–1 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
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 represent 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

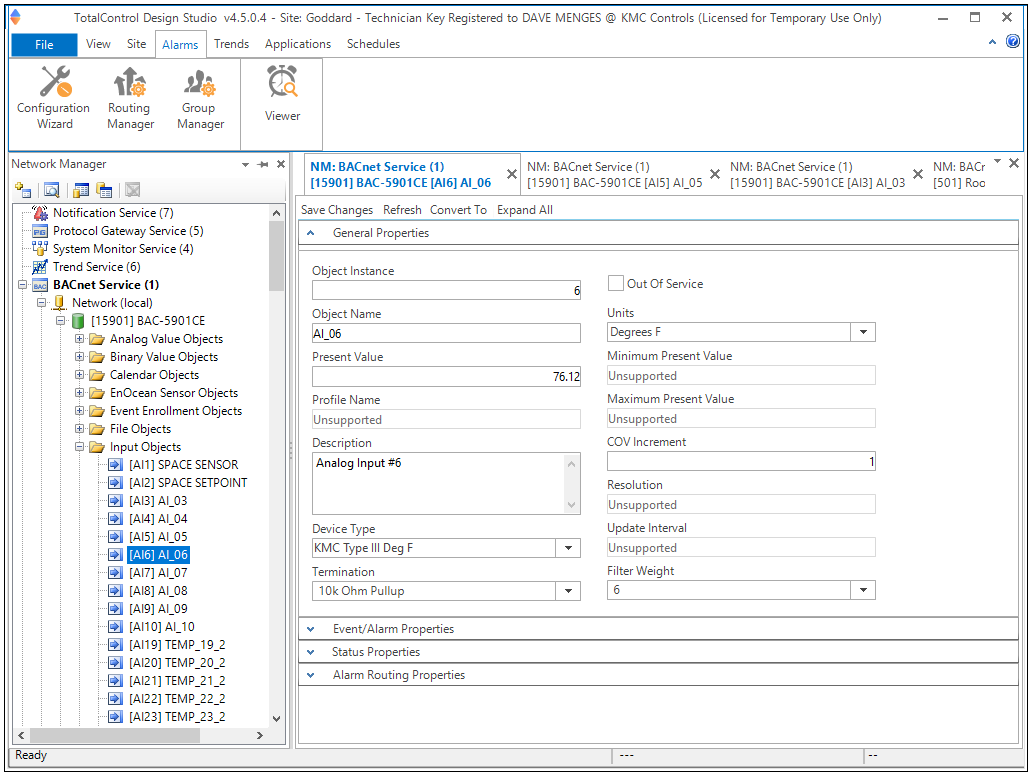
- *BACnet objects and properties* on page 401
- *BACnet services* on page 402

Section 33: Configuring and connecting to BACnet devices and networks

The topics in this section cover connecting to BACnet networks as 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 33–1 Network Manager and Input Configuration tab



- *Adding BACnet devices to the Network Manager list* on page 408
- *Configuring BACnet devices and objects* on page 410
- *Regenerating networks and devices* on page 415
- *Reinitializing a BACnet device* on page 416
- *Setting BACnet system time* on page 417
- *Automatic MAC Addressing commands* on page 418

These additional topics cover more advanced procedures for working with BACnet devices and objects:

- *Reference to BACnet objects* on page 435
- *Configuring BACnet controllers for alarms* on page 135

Adding BACnet devices to the Network Manager list

To add BACnet devices and networks to the Network Manager 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 **All Networks**, the **Local Network**, or a **Specified Network**. When using the Specified Network option, type the network number in the Network ID box.
   - From **Devices**, choose **All Devices**, a **Specified Device**, or a **Range of Devices** specified by the BACnet instance numbers. When using the Specified Device option, type the device instance in the Device Instance box. When using the Range of Devices option, type the minimum device distance in the Min Device Instance box and the maximum device instance in the Max Device Instance box.
3. Click **Find**. The Network Manager list will fill with BACnet networks and devices. See the table *Network Manager BACnet device icons (continued)* on page 410 for a description of the BACnet icons.

Illustration 33–2 The Find BACnet Devices dialog box

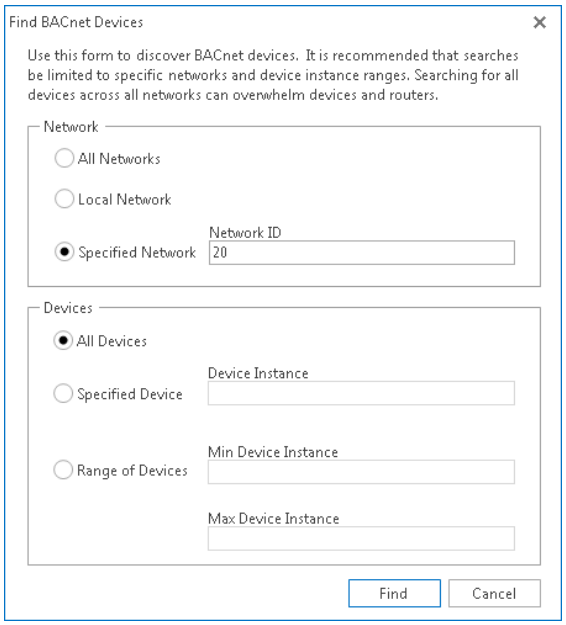


Table 33–1 Network Manager BACnet device icons

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.
	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.

Table 33–1 Network Manager BACnet device icons (continued)

Icon	Status	Action
	Changes have been made to this device that requires a cold start.	See *Reinitializing a BACnet device* on page 416.

Related topics

- *Configuring BACnet devices and objects* on page 410
- *Configuring a BACnet service* on page 413
- *Regenerating networks and devices* on page 415
- *Reinitializing a BACnet device* on page 416
- *Setting BACnet system time* on page 417
- *Automatic MAC Addressing commands* on page 418
- *Using the Network Manager* on page 41
- *Network Manager toolbar commands* on page 42
- *Opening and closing the Network Manager pane* on page 42
- *Reference to BACnet objects* on page 435

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 408. 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.

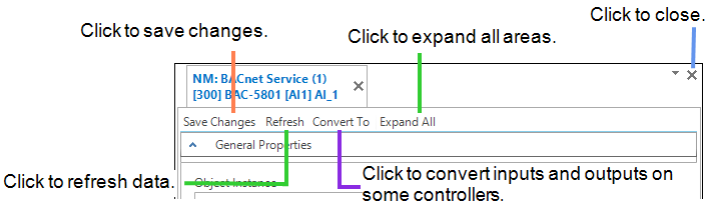
- The device icon will always be in the list under a BACnet service icon and network icon .
- A BACnet object is in a folder under a device. Each type of object has a unique icon.

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 33–3 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 410*.

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 408
- *Configuring BACnet devices and objects* on page 410
- *Regenerating networks and devices* on page 415
- *Reinitializing a BACnet device* on page 416
- *Setting BACnet system time* on page 417
- *Automatic MAC Addressing commands* on page 418

- *Using the Network Manager* on page 41
- *Network Manager toolbar commands* on page 42
- *Opening and closing the Network Manager pane* on page 42
- *Reference to BACnet objects* on page 435
- *Input and output object wizards* on page 673

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 420.

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 408
- *Regenerating networks and devices* on page 415
- *Reinitializing a BACnet device* on page 416
- *Setting BACnet system time* on page 417
- *Input and output object wizards* on page 673
- *Automatic MAC Addressing commands* on page 418
- *Input and output object wizards* on page 673
- *Using the Network Manager* on page 41
- *Reference to BACnet objects* on page 435

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.

Set BACnet Service as Default

The default service in TotalControl can be set to either BACnet PDS, KMD, or OPC. TotalControl graphic elements with a service ID of 0 will use this service.

To set the BACnet Service as the default:

1. Locate the KMD Service icon in the Network Manager list.
2. Right-click the BACnet Service icon and choose **Set as Default** from the shortcut menu.

Note: "Set as Default" is unavailable if the service is already the default.

3. When changes are complete, click **Save Changes to Service** in the tab 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 417.

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 159.

Related topics

- *Opening and closing the Network Manager pane* on page 42
- *Adding BACnet devices to the Network Manager list* on page 408

Regenerating 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 (continued)* on page 410 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 (continued)* on page 410 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 (continued)* on page 410 for a description of icons.

Related topics

- *Adding BACnet devices to the Network Manager list* on page 408
- *Configuring BACnet devices and objects* on page 410
- *Configuring a BACnet service* on page 413
- *Reinitializing a BACnet device* on page 416
- *Setting BACnet system time* on page 417
- *Automatic MAC Addressing commands* on page 418
- *Using the Network Manager* on page 41
- *Network Manager toolbar commands* on page 42
- *Opening and closing the Network Manager pane* on page 42
- *Reference to BACnet objects* on page 435

Reinitializing 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:

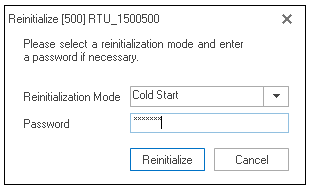
1 In the Network Manager list, click ⊞ or ⊟ to expand or collapse the list to locate a specific device.

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 33–4 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, 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 a BACnet service, network or device.

2 Right-click the service, network or device and choose **Time Synchronization** from the shortcut menu.

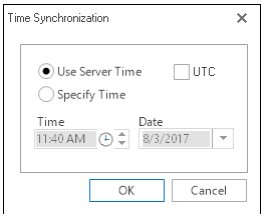
3 Do any of the following:

- 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 413 to set up the BACnet service to automatically update system time.

Illustration 33–5 Time Synchronization dialog



Related topics

- *Regenerating networks and devices* on page 415
- *Adding BACnet devices to the Network Manager list* on page 408
- *Configuring BACnet devices and objects* on page 410
- *Regenerating networks and devices* on page 415
- *Reinitializing a BACnet device* on page 416
- *Automatic MAC Addressing commands* on page 418
- *Using the Network Manager* on page 41
- *Network Manager toolbar commands* on page 42
- *Opening and closing the Network Manager pane* on page 42
- *Reference to BACnet objects* on page 435

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 one of the following commands from the shortcut menu.

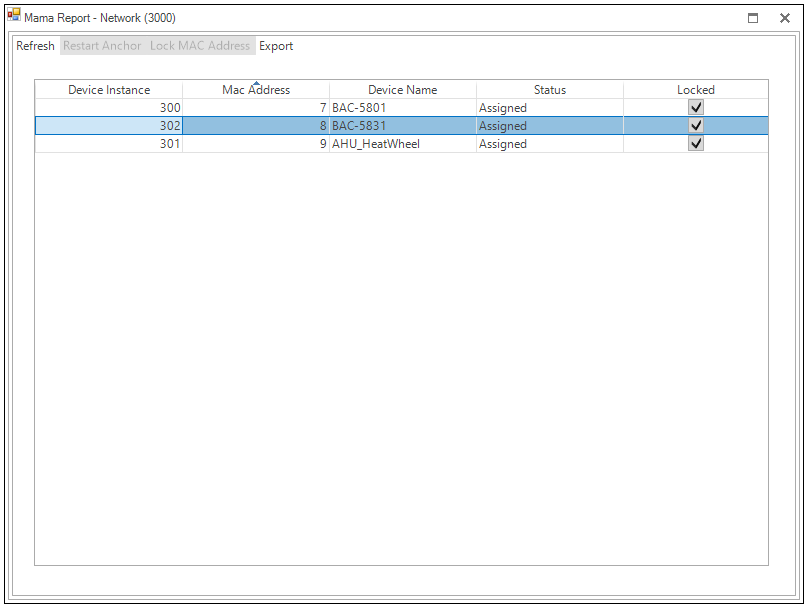
Request Status Displays the status of automatic addressing in the controller in the output window.

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.

MAMA Report This command generates a report that shows the status of all controllers on the network with MS/TP Automatic MAC Addressing. The report can be exported to a `.csv` file.

Illustration 33–6 MAMA report



Related topics

- *Adding BACnet devices to the Network Manager list* on page 408
- *Configuring BACnet devices and objects* on page 410
- *Regenerating networks and devices* on page 415
- *Reinitializing a BACnet device* on page 416
- *Setting BACnet system time* on page 417
- *Using the Network Manager* on page 41
- *Network Manager toolbar commands* on page 42
- *Opening and closing the Network Manager pane* on page 42
- *Reference to BACnet objects* on page 435

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 its 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.

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 412.



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 408
- *Regenerating networks and devices* on page 415
- *Reinitializing a BACnet device* on page 416
- *Setting BACnet system time* on page 417
- *Automatic MAC Addressing commands* on page 418
- *Input and output object wizards* on page 673
- *Using the Network Manager* on page 41
- *Reference to BACnet objects* on page 435

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 series 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 *Reinitializing a BACnet device* on page 416.

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.

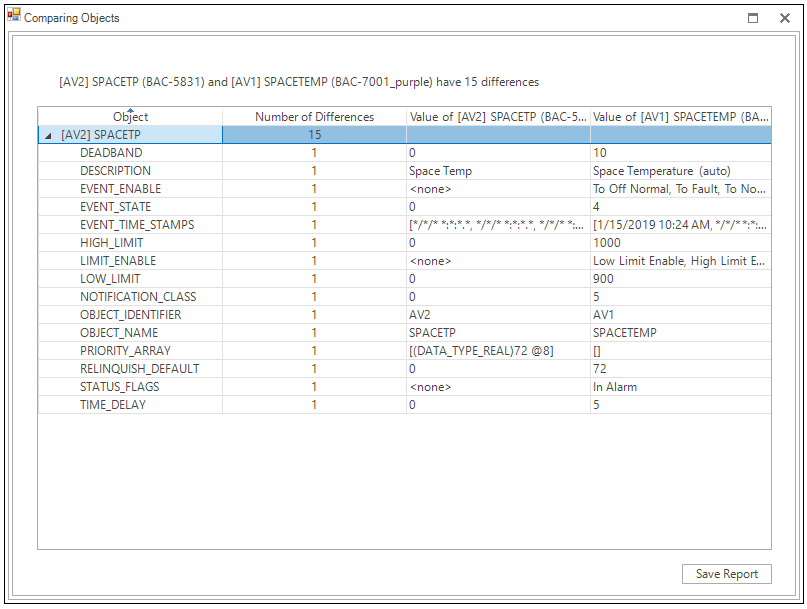
Comparing BACnet objects

BACnet objects in either the Resource Manager or Network Manager can be compared with each other.

1. Open the Network Manager, Resource Manager, or both.
2. Reveal the objects for comparison.
3. Press and hold **Shift**.
4. Drag one object and drop it on a second object.

   Once an object is dropped onto another object, KMC Connect compares the properties in the two objects and generates a report.
5. Do either of the following:
   - Click the triangle ▷ next to the object name to reveal the report.
   - Click **Save Report** to save the report as `.csv` file.

Illustration 33–7 Report from comparing two objects



Section 34: Backing up and restoring BACnet and KMDigital devices

This section describes the methods to back up BACnet and KMDigital devices with KMC Connect.

KMC Connect supports several methods to back up BACnet devices.

- *Backing up with .bnd files* on page 425 describes saving a `.bnd` file. A `.bnd` file is the only file type that can be edited with Design Studio.
- *BACnet backup and restore* on page 426 covers the procedures to archive device configurations from other manufacturers.
- *Importing and exporting BACstage files* on page 428 describes opening and saving `.bac` files that are compatible with the BACstage operator workstation.
- *Scheduling BACnet and KMDigital devices for backup* on page 429 details the procedures to back up BACnet and KMDigital devices on a regular basis.

Backing up with `.bnd` files

To create `.bnd` backup files, drag items from the Network Manager list to the Resource Manager pane. A `.bnd` file is the only file type that can be edited with Design Studio. KMC Connect will back up any of the following items with this method.

- A single device
- All or some of the devices from a network or service
- A single object
- All of the objects in a group folder within a device

To restore devices or objects with a `.bnd` file, see the topic *Restoring a configuration* on page 59.



Note: A `.bnd` file is the only file type that can be edited with KMC Connect.

To save device configurations to `.bnd` files, do the following:

1. Open the Network Manager.
2. Expand or collapse the Network Manage list to locate a BACnet device , network , or service  icon.
3. Drag the icon to the Resource Manager pane.
4. For BACnet networks or services, the Export Devices dialog opens. To delete a device from the backup process, clear the check box next to the device name.

5 Click **OK** to continue.

To save object configurations to `.bnd` files, do the following:

1 Open the Network Manager.

2 Expand or collapse the Network Manage list to locate the BACnet device.

3 Expand the device to locate the object folders.

4 Do either of the following:

- Expand a folder and drag a single object to the Resource Manager.
- Drag the folder to the resource manager.

To export a `.bnd` file, do the following:

Exporting to a `.bnd` file is the same as dragging an item to the Resource Manager except that the backup files can be saved to any location available to the computer running KMC Connect.

1 Open the Network Manager.

2 Expand or collapse the Network Manage list to locate a BACnet device, network, or service icon.

3 Do one of the following:

- Right-click a device icon and choose **Export Configuration** from the shortcut menu.
- Right-click a network or service icon and choose **Export Device Configurations** from the shortcut menu.

4 For BACnet networks or services, the Export Devices dialog opens. To delete a device from the backup process, clear the check box next to the device name.

5 Select a location to store the backup files.

6 Click **Okay** to start the export.

Related topics

- *Restoring a configuration* on page 59
- *BACnet backup and restore* on page 426
- *Importing and exporting BACstage files* on page 428
- *Scheduling BACnet and KMDigital devices for backup* on page 429
- *About the Resource Manager* on page 57

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 manufacturer's 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.

A `.bbk` file is unique to KMC Controls, Inc. and cannot be used by programs from other manufacturers. It cannot be edited by KMC Connect.

Preparation Time The Preparation Time is a delay built into KMC Connect 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 the Network Manager.
2. Expand or collapse the Network Manage list to locate a BACnet device, network, or service icon.
3. Right-click the icon and choose **BACnet Backup** from the shortcut menu.
4. Select or clear the check boxes next to the name of each device.
5. Select a location to store the backup file. The default location is the BACnetBackup folder in the Resource Manager. KMC Connect will automatically create the folder if it is not present.
6. Click **Save** to start the export.

To restore a device from a `.bbk` file, do the following:

1. Open the Network Manager.
2. Expand or collapse the Network Manage list to locate a BACnet device icon.
3. Right-click the device icon and choose **BACnet Restore** from the shortcut menu.
4. If required, enter a password and then click **Restore**.
5. Locate the `.bbk` file.
6. Click **Open** to start the import.

Related topics

- *Backing up with .bnd files* on page 425
- *Importing and exporting BACstage files* on page 428
- *Scheduling BACnet and KMDigital devices for backup* on page 429
- *About the Resource Manager* on page 57

Importing and exporting BACstage files

Use the Resource Manager to import BACstage `.bac` configuration files. Once a `.bac` file is converted to a `.bnd` file, it can be configured or used the same as any other `.bnd` file. For example, you may drag an input object from the `.bnd` file to a BACnet input object in the Network Manager list.

KMC Connect will also convert `.bnd` files into `.bac` files.

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. Right-click the `.bac` file and choose **Save as .bnd** from the shortcut menu.

To export a BACstage `.bac` file

1. Locate a `.bnd` file in the Resource Manager list.
2. Right-click the file and choose **Save as .bac** from the shortcut menu.
3. Select the file type and then click **OK**.

Related topics

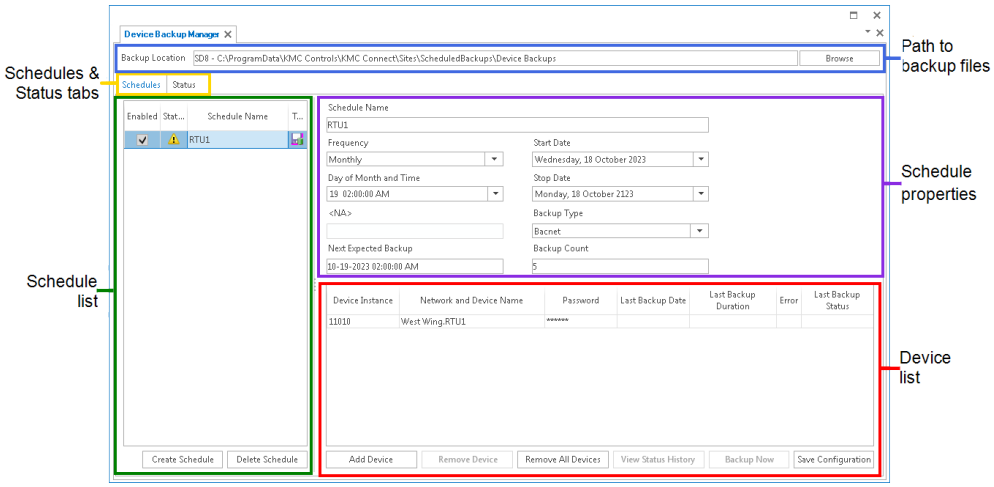
- *BACnet backup and restore* on page 426
- *Backing up with .bnd files* on page 425
- *Scheduling BACnet and KMDigital devices for backup* on page 429
- *About the Resource Manager* on page 57

Scheduling BACnet and KMDigital devices for backup

Use the Device Backup Manager to back up specific devices on a regular schedule. The Device Backup Manager operates with the following rules:

- Scheduled backups run on the date and time as maintained in the computer running KMC Connect.
- Devices can be added to more than one schedule.
- If the same devices are in overlapping schedules of the same type, files are saved for the first schedule and then copies are saved for each additional schedule.
- If the same devices are in overlapping schedules of different types, files for each device are saved only for the first schedule.
- A failed backup will not be retried.
- Disabling or deleting a backup in progress will stop the backup for all remaining devices. Existing device files will not be deleted or modified.
- If a backup is scheduled for a day of the month that does not exist, the backup will take place on the last day of the month.

Illustration 34–1 Device Backup Manager - Schedules dialog box



Opening the Device Backup Manager

The Device Backup Manager can be opened from either the ribbon or the Network Manager.

From the Ribbon On the ribbon, choose **Device Backup Manager** from the **Site** group.

From the Network Manager From the Network Manager, do the following:

1 Open the Network Manager.

2 Locate the System Monitor icon .

3 Right-click the icon and choose **Device Backup Manager** from the shortcut list.

Note: Attempting to open the Device Backup Manager after it is already open brings the existing form to the front.

Creating new schedules

To set up a new schedule, do the following:

1 Open the **Device Backup Manager**.

2 Do one of the following:

- In the Device Backup Manager dialog, click **Create Schedule**.
- Right-click the Schedule List and choose **Create from shortcut** list.

3 Set the schedule properties.

- In **Frequency**, set how often the scheduled backup will take place.
- In **Time, Day of Week and Time, Day of Month and Time**, or **Date and Time**, set the time of day the backup will take place.
- In **Start Date** and **End Date**, set the dates for the backups to occur.
- In **Backup Type**, choose the type of backup (Bnd, Bacnet, or Kmd).

4 Add devices to the Device list by doing one of the following:

- Drag a single device icon from the Network Manager.
- Drag a network icon or service icon to add all devices from a network or service.
- Click **Add Device** and search for devices as described in *Searching for devices*.

Note: BACnet devices (`.bnd` and `.bbk` backup file types) and KMD devices (`.kmd` backup file type) cannot be used in the same schedule.

5 Click **Save Configuration** when finished.

Changing existing schedules

To modify an existing schedule, do the following:

1 Open the **Device Backup Manager**.

2 Choose a schedule from the Schedules list.

3 As required, make changes to the Schedule properties or the Device list.

4 Click **Save Configuration** when finished.

Searching for devices

Objects in the Network Manager list can be added to the Device list by searching for a device instance, a device name, or a series of devices with a pattern to their names.

To add devices by searching, do the following:

1. Create a new schedule or select an existing schedule.
2. Under the Device list, click **Add Device**. The Search dialog opens.
3. In **Device**, enter the instance number or name of a device.

   For devices with a pattern to their names, use an asterisk (*) or question mark (?) to replace characters.

   - Use an asterisk (*) to replace a group of characters. *Room1** will find *Room10* and *Room111* but not *Room20*.
   - Use a question mark (?) to replace a single character. *Room150?* will find *Room1501*and *Room1502* but not *Room15* or *Room150*.
4. Click **Search**.
5. Select or clear the check box next to each device found by the search.
6. Click **OK** to add the selected devices to the Device list.

Illustration 34–2 Find BACnet Devices dialog box

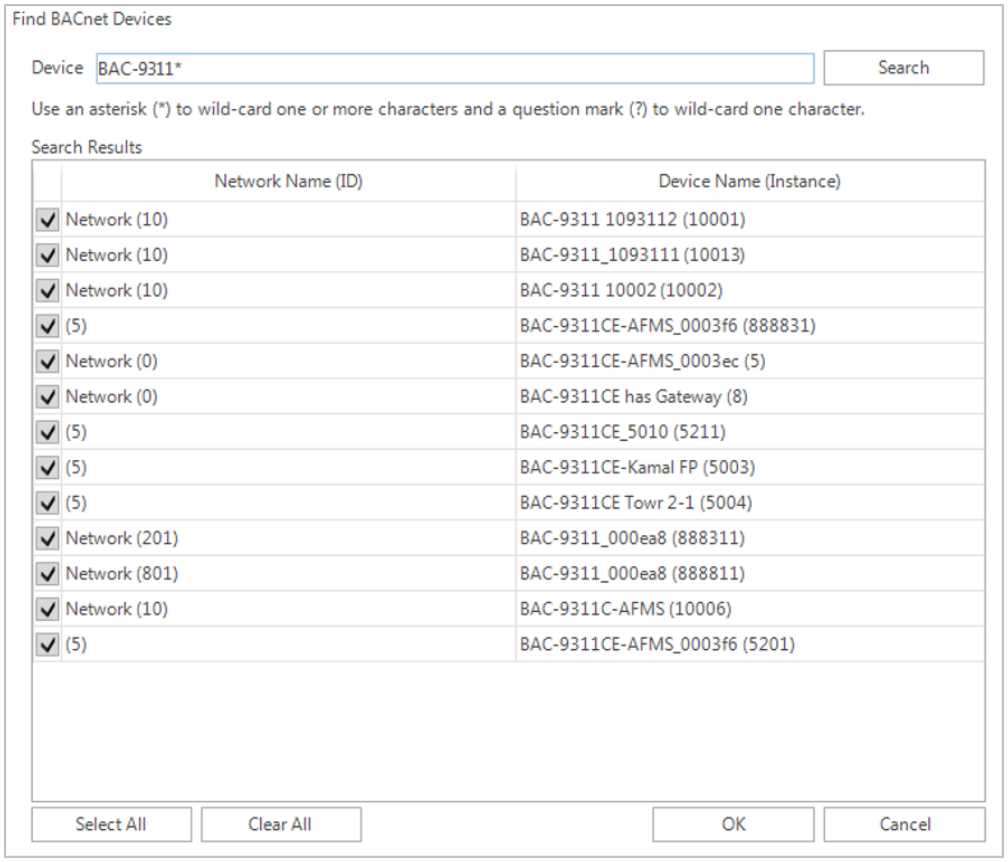
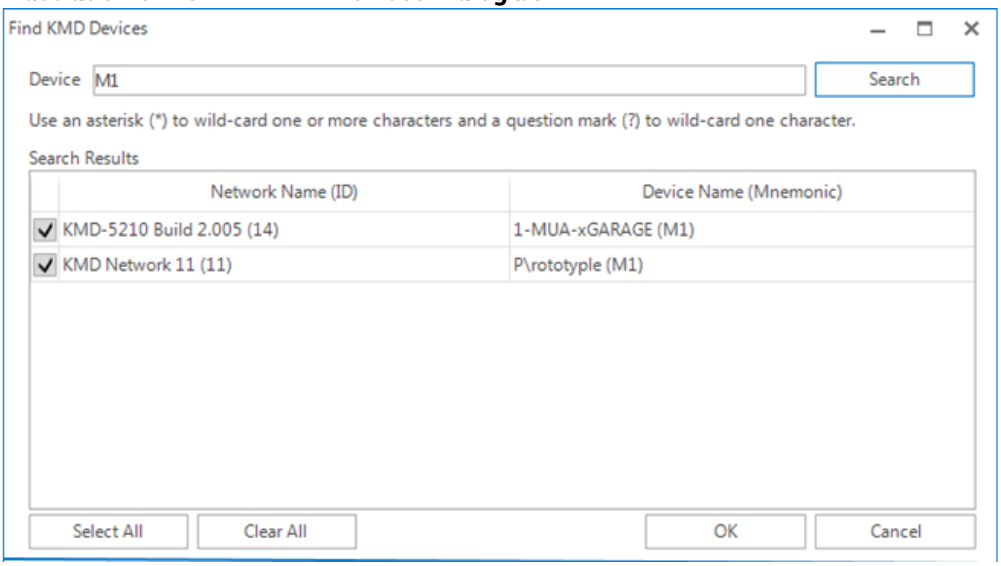


Illustration 34–3 Find KMD Devices dialog box



Properties in the Schedules tab

Status Icons The Status Icons in the Schedule List indicate the state or progress of each schedule. Additional information about the schedule may be listed in the Output Window.

Table 34–1 Backup Status Icons

	Backup scheduled and waiting to start
	Backup running as scheduled
	Backup complete
	Backup canceled

Backup Location This is the path to the storage location for the device backup files. The location is always on the computer running KMC Connect and cannot be changed.

Browse Click to open a dialog box used to select the folder containing the backup files.

Frequency This property sets the interval for scheduled backups.

- Custom—The scheduled backup runs at an interval of one or more hours between backups. The interval is set in **Custom Period Hours**.
- Every Midnight—The backup runs daily at midnight.
- Daily—The backup runs once a day as designated in **Time**.
- Weekly—The backup runs once a week as designated in **Day of Week and Time**.
- Biweekly—The backup runs every two weeks as designated in **Day of Week and Time**.
- Monthly—The backup runs once a month as designated in **Day of Month and Time**.
- One Time—Runs a single backup as designated in **Date and Time**.

Only one of the properties in the following group are active at one time. The active property depends on the setting of Frequency.

Time Sets the time of day for a Daily schedule. It is active only when Frequency is set to Daily.

Date and Time Sets the date and time for a One Time backup. It is active only when Frequency is set to One Time.

Day of Week and Time Sets the day of the week and time of day for a Weekly or Biweekly schedule. It is active only when Frequency is set to Weekly or Biweekly.

Day of Month and Time Sets the day of the month and time of day for a Monthly schedule. It is active only when Frequency is set to Monthly.

Custom Period Hours Sets the interval for a Custom backup interval. It is active only when Frequency is set to Custom.

Start Date and Stop Date A scheduled backup starts only between midnight on the Start Date and midnight on the Stop Date.

Backup type The Device Backup Manager can save the following types of backup files.

- BACnet—This is a standard BACnet backup file. The file extension is `.bbk`.
- BND—A KMC Connect BACnet `.bnd` file that can be edited in the Resource Manager.
- KMDigital—A `.kmd` file that can be edited in the Resource Manager.

Properties in the Status tab

The **Devices with Backups** portion of the Status tab displays at the top half of the screen and lists all devices that are scheduled for backup.

Device Instance The BACnet device instance or the KMD Device ID.

Network and Device Name Shows the device's network and device name in the format <network name>.<device name>.

Last Backup Date The date of the most recent backup for the device.

Last Backup Duration The length of time taken to create the newest backup of this device. This field is blank if there has been no backup.

Schedule Name The name of the schedule that created the newest backup, or, if no backup has been made, the name of the schedule that will create the next backup.

Last Backup Status The status of the last backup that was executed for the device; blank if there are no backups.

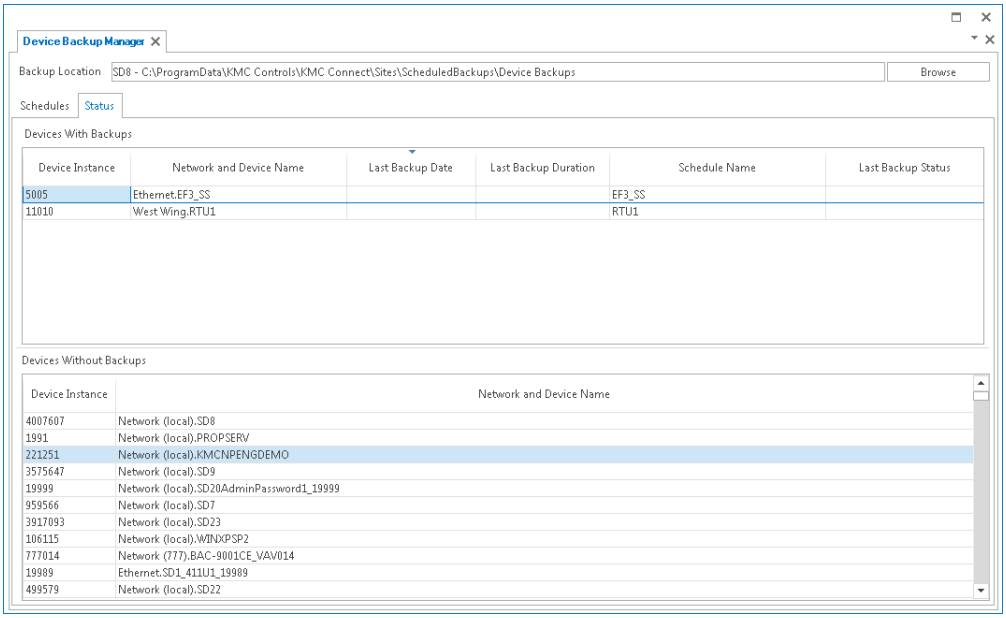
The **Devices without Backups** portion of the Status tab displays at the bottom half of the screen and lists all devices that are not included in a backup schedule.

Listed devices that are not part of any backup can be selected and added to a new backup schedule.

Device Instance The BACnet device instance or the KMD Device ID.

Network and Device Name Shows the device's network and device name in the format <network name>.<device name>.

Illustration 34–4 Device Backup Manager - Status dialog box



Section 35: 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 the Network Manager* on page 41.
- To configure the objects and properties in a BACnet device see *Configuring BACnet devices and objects* on page 410.
- For an introduction to BACnet see *BACnet overview* on page 401.

This section describes the following BACnet objects:

- *Access Door objects* on page 435
- *Averaging objects* on page 440
- *Calendar objects* on page 441
- *Command object* on page 442
- *Device objects* on page 443
- *Event enrollment objects* on page 451
- *Event Log objects* on page 455
- *File objects* on page 457
- *Group objects* on page 457
- *Input objects-accumulator* on page 458
- *Input objects-analog* on page 461
- *Input objects-binary* on page 466
- *Input objects-multistate* on page 469
- *Life Safety Point objects* on page 472
- *Life Safety Zone objects* on page 476
- *Loop objects* on page 484
- *Notification objects* on page 490
- *Output objects-analog* on page 491
- *Output objects-binary* on page 494
- *Output objects-multistate* on page 498
- *Schedule objects* on page 504
- *Trend Log objects* on page 509
- *Trend Log Multiple objects* on page 514
- *Value objects-analog* on page 518
- *Value objects-binary* on page 521
- *Value objects-multistate* on page 525

Access Door objects

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 401.

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. Printable characters must be used for the Name property.

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 402.

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 requirements for this object. The definition of the profile is unique to the vendor 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 402.

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 143
- *About intrinsic BACnet alarms and events* on page 139
- *Notification objects* on page 490

Notification Class Specifies the notification class object to be used when handling and generating event notifications for this object. See *Notification objects* on page 490 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 Too Long
- Forced Open
- Tamper
- Door Fault

Fault Values The Door Alarm State property must equal one of the selected items in the Alarm Values property list before a To Fault event is generated.

- Normal
- Alarm
- Door Open Too 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 in 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 35–1 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.

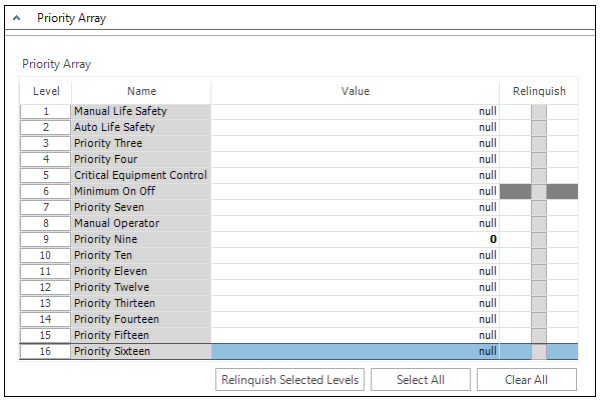
Priority Array Properties

The Priority Array property holds a priority list for writing values for the object.

To relinquish a priority value, select the check box and then click **Relinquish**.

For an explanation, see the topic *Priority arrays* on page 402.

Illustration 35–1 Priority Array Properties



Alarm Routing Properties

Alarm classes designate a group of operators who receive notifications of alarms and events originating from this object. See the topics *An overview of notifications, alarms, and events* on page 133 and *Emailing operator alarms* on page 171.

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 410.

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. Printable characters must be used for the Name property.

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 requirements for this object. The definition of the profile is unique to the vendor 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**— The **Local** device within which the object to be averaged is located.
- **Object**— Select an Analog object that exists on the local BACnet device.
- **Property**–Select the property to be evaluated 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 the 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 objects

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 objects* on page 504 for more information about setting up schedules.
- To configure the properties in a BACnet device, see *Configuring BACnet devices and objects* on page 410.

Object Instance The calendar object number. Calendar 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. Printable characters must be used for the Name property.

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 of the vendor's object profile to which this object conforms. A profile name defines a set of additional properties, behaviors and requirements for this object. The definition of the profile is unique to the vendor 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, 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 & 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, then `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 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 410.

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. Printable characters must be used for the Name property.

Present Value The Present Value property commands which of the action lists 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 of the vendor's object profile to which this object conforms. A profile name defines a set of additional properties, behaviors and requirements for this object. The definition of the profile is unique to the vendor 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 the general characteristics of the user interface in a KMC 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 KMC BACnet controller.

To configure the objects and properties in a KMC BACnet controller, see the topic *Configuring BACnet devices and objects* on page 410.

General Properties

Device Instance The device instance uniquely identifies the device on the internetwork and 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 KMC 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 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 KMC Conquest controllers only) Enter the new MS/TP 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 requirements for this object. The definition of the profile is unique to the vendor 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 maintained in 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 maintained in 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 minutes. 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.

IP Configuration Properties

See the topic *Configuring the IP connection* on page 421.

Communication Properties

MAX APDU Size This property is the maximum number of octets that may be contained in a single, indivisible APDU (Application Layer protocol Data Unit)[1].

APDU Timeout Indicates the period — in milliseconds — between retransmissions of an APDU that requires 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 (For KMC Conquest controllers only) 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. See *KMC Conquest-compliant controllers* on page 709.

[1]Application Layer Protocol Data Unit. An APDU is the significant data in a network packet.

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 MS/TP 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 35–2 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.

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 KMC Connect.

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 *KMC Conquest-compliant controllers* on page 709.

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 identification number assigned to the deployed application. This is empty if an application has not been deployed.

NFC Properties

NFC (Near Field Communication) properties are unique to the Conquest series of controllers. Near Field Communication for an individual controller can be disabled or enabled with the NFC commands.



Note: Use only the Enable NFC and Disable NFC commands. Other commands should be used only under the direction of technical support from KMC Controls, Inc.

To enable or disable the NFC in a controller, do the following:

1. From the Network Manager list, locate a Device object icon .
2. Double-click the icon to open the configuration pane.
3. Expand the NFC Properties area.
4. From the **Direct Command** list, choose either **Enable NFC** or **Disable NFC**.
5. At the top of the pane, click **Save Changes**.

To disable all of the controllers on a BACnet service or network, see the topic *Enable and disable NFC* on page 52.

Object Locking

The properties in Object Locking hide and lock the Control Basic programs in a BACnet Program object. Locked files cannot be viewed or edited with the Code Editor, but properties in the Program object are still accessible.

Password The password required to lock and unlock a program.

Locked Objects A list of programs that are locked and hidden from view in Program objects. See the topic *Locking Control Basic programs* on page 211.

Niagara Compatibility Mode

Event State Override No longer supported.

Priority NULL Override When selected, the controller is set to respond to the Niagara Set command. In this mode, when the controller receives a BACnet write with a non-NULL value and no priority level is designated, it will write the value only to the relinquish default property. It will not write to priority level 16 as would take place for a normal BACnet write.

Also in this mode, a write with a value of NULL and no priority will be acknowledged with a Simple ACK and the controller will drop the write, relinquish default will remain unchanged, and nothing will happen with priority 16.

To write a value at priority level 16 in compatibility mode, priority level 16 must be specified for the write.

Display objects

The Display object is a proprietary BACnet object that represents the general characteristics of the user interface in a KMC Conquest FlexStat. There is only one Display object in each Conquest FlexStat.

Use the Display Objects configuration page in KMC Connect to set the Display object properties in a KMC Conquest FlexStat. See the topic *Configuring Conquest FlexStats* on page 551.

Event enrollment objects

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 (continued).

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. Printable characters must be used for the Name property.

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 35–3 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

Table 35–3 Event Types, Event States, and Event Parameters (continued)

Event Type	Event State	Event Parameters
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
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 objects* on page 490
- *Configuring BACnet controllers for alarms* on page 135

Notification Class Specifies the notification class object to be used when handling and generating event notifications for this object. See *Notification objects* on page 490 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

Alarm classes designate a group of operators who receive notifications of alarms and events originating from this object. See the topics *An overview of notifications, alarms, and events* on page 133 and *Emailing operator alarms* on page 171.

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 410.

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. Printable characters must be used for the Name property.

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 requirements for this object. The definition of the profile is unique to the vendor 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 objects* on page 490 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 35–4 Event log 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 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

Alarm classes designate a group of operators who receive notifications of alarms and events originating from this object. See the topics *An overview of notifications, alarms, and events* on page 133 and *Emailing operator alarms* on page 171.

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 410.

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. Printable characters must be used for the Name property.

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 requirements for this object. The definition of the profile is unique to the vendor 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 objects–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 410.

Related topics

- *BACnet objects and properties* on page 401
- *Input objects-analog* on page 461
- *Input objects-binary* on page 466
- *Input objects-multistate* on page 469

General Properties

Object Instance The number of the object. Input objects are numbered sequentially within the 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. Printable characters must be used for the Name property.

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 of the vendor's object profile to which this object conforms. A profile name defines a set of additional properties, behaviors and requirements for this object. The definition of the profile is unique to the vendor 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 565 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 objects* on page 490 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 35–5 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)
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

Alarm classes designate a group of operators who receive notifications of alarms and events originating from this object. See the topics *An overview of notifications, alarms, and events* on page 133 and *Emailing operator alarms* on page 171.

Input objects–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 673
- *Configuring BACnet devices and objects* on page 410
- *Calibration* on page 463

Related topics

- *BACnet objects and properties* on page 401
- *Input objects-accumulator* on page 458
- *Input objects-binary* on page 466
- *Input objects-multistate* on page 469

General Properties

Object Instance The input object number. Input objects are numbered sequentially within the 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. Printable characters must be used for the Name property.

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 of the vendor's object profile to which this object conforms. A profile name defines a set of additional properties, behaviors and requirements for this object. The definition of the profile is unique to the vendor 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 565 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 35–6 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

To add a calibration factor to a BACnet analog input object, do the following:

6 Open the configuration tab for the object.

7 Right-click **Present Value**.

8 Choose **Calibrate Present Value** from the shortcut menu.

9 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

Notification Class Specifies the notification class object to be used when handling and generating event notifications for this object. See *Notification objects* on page 490 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 35–7 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)
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

Alarm classes designate a group of operators who receive notifications of alarms and events originating from this object. See the topics *An overview of notifications, alarms, and events* on page 133 and *Emailing operator alarms* on page 171.

Input objects–binary

A 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 input object types.

To configure a property, see the topics *Input and output object wizards* on page 673 and *Configuring BACnet devices and objects* on page 410.

Related topics

- *BACnet objects and properties* on page 401
- *Input objects-accumulator* on page 458
- *Input objects-analog* on page 461
- *Input objects-multistate* on page 469

General Properties

Object Instance The input object number. Input objects are numbered sequentially within the 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. Printable characters must be used for the Name property.

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 of the vendor's object profile to which this object conforms. A profile name defines a set of additional properties, behaviors and requirements for this object. The definition of the profile is unique to the vendor 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 then 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 35–8 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 objects* on page 490 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 35–9 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)
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

Alarm classes designate a group of operators who receive notifications of alarms and events originating from this object. See the topics *An overview of notifications, alarms, and events* on page 133 and *Emailing operator alarms* on page 171.

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, 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 410.

Related topics

- *BACnet objects and properties* on page 401
- *Input objects-accumulator* on page 458
- *Input objects-analog* on page 461
- *Input objects-binary* on page 466

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. Printable characters must be used for the Name property.

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 of the vendor's object profile to which this object conforms. A profile name defines a set of additional properties, behaviors and requirements for this object. The definition of the profile is unique to the vendor 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 objects* on page 490 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 35–10 Multistate 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)
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

Alarm classes designate a group of operators who receive notifications of alarms and events originating from this object. See the topics *An overview of notifications, alarms, and events* on page 133 and *Emailing operator alarms* on page 171.

Life Safety Point objects

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 objects* on page 476.

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. Printable characters must be used for the Name property.

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 of the vendor's object profile to which this object conforms. A profile name defines a set of additional properties, behaviors and requirements for this object. The definition of the profile is unique to the vendor 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 objects* on page 490 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, 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 35–11 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:

- 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 objects

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 objects* on page 472.

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. Printable characters must be used for the Name property.

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 of the vendor's object profile to which this object conforms. A profile name defines a set of additional properties, behaviors and requirements for this object. The definition of the profile is unique to the vendor 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 objects* on page 490 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, 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 35–12 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. Printable characters must be used for the Name property.

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 of the vendor's object profile to which this object conforms. A profile name defines a set of additional properties, behaviors and requirements for this object. The definition of the profile is unique to the vendor 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 35–13 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 objects* on page 490 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.

Load Control 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	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

Alarm classes designate a group of operators who receive notifications of alarms and events originating from this object. See the topics *An overview of notifications, alarms, and events* on page 133 and *Emailing operator alarms* on page 171.

Loop objects

Use the Loop objects dialog to configure a PID loop. A PID loop is a mathematical function that calculates an analog output required to maintain a process at or near a setpoint. The output of the loop object directly controls a property—usually the present value property—of an object.

To program the properties of a loop object with Control Basic, see *Mnemonics for BACnet controllers* on page 227.

To use Autotune, see *Autotune Properties* on page 488.

General Properties

Object Instance The loop object number. Loop 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. Printable characters must be used for the Name property.

Present Value A numerical property that indicates the current value—in engineering units—of the output of the loop object.

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 requirements for this object. The definition of the profile is unique to the vendor 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 the 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 565.

Reference Properties

Action The action of the loop. Action can be either direct acting or reverse acting.

- A Direct acting loop object increases the value of the Manipulated Variable Reference property as the value of the Controlled Variable Reference property rises above the value of the Setpoint property. Typically, direct action loops are used for cooling systems.
- A Reverse acting loop objects decreases the value of the Manipulated Variable Reference property as the value of the Controlled Variable Reference property rises above the value of the Setpoint property. Typically, reverse action loops are used for heating systems.

Controlled Variable Reference (Input) The controlled variable is usually the present value of an object that is measuring a process variable such as a temperature, airflow, or pressure. 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.

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 is entered in the Setpoint property text box.
- 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.

Setpoint A fixed value for Setpoint Reference when Setpoint Reference is set to None.

Manipulated Variable Reference (Output) The Manipulated Variable Reference property directs the output of the loop to a property within 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 the Present Value property–within the referenced object.

Priority For Writing Sets the priority for writing to the object in Manipulated Variable Reference. See the topic *Priority arrays* on page 402.

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.

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.

Update Interval Indicates the interval—expressed in milliseconds—that the loop algorithm updates the Present Value property of the loop object.

Note: For BACnet controllers from KMC Connect, the Update Interval property is fixed at 1,000 milliseconds and cannot be changed. The 1,000 millisecond interval is also the calculation rate in controllers from KMC Connect.

COV Increment This property specifies the minimum change of Present Value that will send a COV notification to subscriber notification clients.

Constant Properties

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%.

In general, increasing the proportional constant increases the response speed of the control system

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 additonal 15% would be applied over the couse of 1 minute in a direction to reduce the rate of change, 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 a derivative is not used.

Event/Alarm Properties

Notification Class Specifies the notification class object to be used when handling and generating event notifications for this object. See *Notification objects* on page 490 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 the 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 35–14 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.

For more infomation on PID loops, see Application Guide AG190614, PID Loops in KMC controllers.

Autotune Properties

When active, the Autotune algorithm calculates values for the loop's proportional, integral, and derivative constants. The algorithm calculates the constants by operating the loop in an installed and configured controller and then measuring the response. The Autotune algorithm uses the Ziegler-Nichols method to calculate the constants and is unique to BACnet controllers from KMC Controls.

Note: Autotune is available only in Conquest controllers with firmware R1.2.0.9 or later.

The Autotune process The algorithm starts by setting the Proportional constant to a high value and setting the Derivative and Integral constants to zero. This purposely makes the loop unresponsive. Then, the Proportional constant is slowly decreased until the loop's present value begins oscillating–or hunting–around the setpoint. Once oscillation is detected, the algorithm measures the period of oscillation. When a valid measurement is acquired, the algorithm stops monitoring and saves the Proportional constant value from the time that the oscillation measurement was acquired. It then calculates and saves new Derivative and Integral constants based on the period of the oscillation.

Conditions to consider for using the Autotune algorithm:

- The Controlled Variable Reference property must be able to reach setpoint. If not, tuning will never finish.
- The loop must be actively controlling the output and the environment it is controlling must be left relatively undisturbed during the Autotune process.
- The Autotune algorithm has a built-in restart. If tuning is greatly disturbed, the processwill start over.
- If the controller is restarted, tuning will start over.

To use Autotune, do the following:

1. Configure the loop properties. The properties are describe in the sections *General Properties* on page 484, *Constant Properties* on page 486, and *Reference Properties* on page 484.
2. Choose a **Tuning Type**.
3. Optional: Monitor progress in a Trend Viewer window by doing the following:
   a. Open a Trend Viewer window.
   b. Add the Controlled Variable Reference and Setpoint Reference properties to the viewer.
4. Select the **Enable Autotune** check box.
5. Click **Save Changes** at the top of the tab.

Autotune Properties The following properties control the Autotune algorithm.

Enable Autotune When selected, the Autotune algorithm runs in the controller.

Autotune State This property displays the progress of the Autotune algorithm.

- **Idle** The Autotune algorithm is monitoring the loop output for oscillation around the setpoint.
- **Achieving Oscillation** The algorithm has detected an oscillation in the output of the loop.
- **Measure Oscillation** The algorithm is measuring the period of oscillation.
- **Calculating Constants** The algorithm is calculating the Derivative and Integral constants based on the oscillation time and saving the Proportional constant from the time when the oscillation was measured.

Tuning Type This property determines which of the PID constants are calculated by the Autotune algorithm:

- Proportional Only
- Proportional and Integral
- Proportional, Integral and Derivative

Alarm Routing Properties

Alarm classes designate a group of operators who receive notifications of alarms and events originating from this object. See the topics *An overview of notifications, alarms, and events* on page 133 and *Emailing operator alarms* on page 171.

Notification objects

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. Printable characters must be used for the Name property.

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 requirements for this object. The definition of the profile is unique to the vendor 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 35–15 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 401
- *Priority arrays* on page 402
- *Output objects-binary* on page 494
- *Analog output object wizard* on page 677

General Properties

Output Instance

The output object number. Output objects are numbered sequentially within the 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 402.

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 requirements for this object. The definition of the profile is unique to the vendor 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 402.

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 565. 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 494 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 objects* on page 490 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 35–16 Output 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 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.

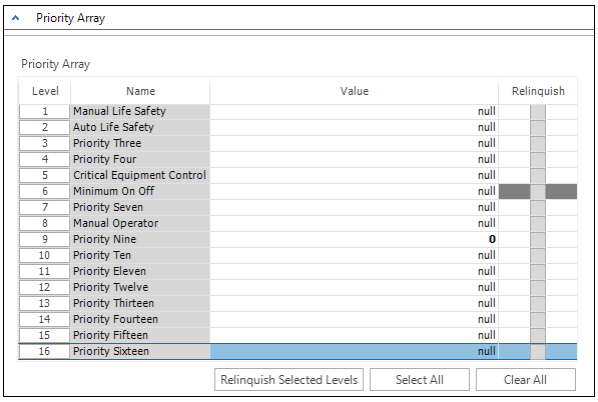
Priority Array Properties

The Priority Array property holds a priority list for writing values for the object.

To relinquish a priority value, select the check box and then click **Relinquish**.

For an explanation, see the topic *Priority arrays* on page 402.

Illustration 35–2 Priority Array Properties



Alarm Routing Properties

Alarm classes designate a group of operators who receive notifications of alarms and events originating from this object. See the topics *An overview of notifications, alarms, and events* on page 133 and *Emailing operator alarms* on page 171.

Output objects–binary

A binary output is a standard BACnet object. The properties in the object set the characteristics of a 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 401
- *Output objects-analog* on page 491.
- *Priority arrays* on page 402

General Properties

Object Instance The output object number. Output objects are numbered sequentially within the 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. Printable characters must be used for the Name property.

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 495.

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 402.

Write Priority When saving the object, this property sets priority for the Present Value property. See the topic *Priority arrays* on page 402.

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 requirements for this object. The definition of the profile is unique to the vendor 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 402.

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 35–17 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 objects* on page 490 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 402.

Status Properties

Status Flags The four BACnet status flags are an indication of the general condition of the object.

Table 35–18 Binary output 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 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).

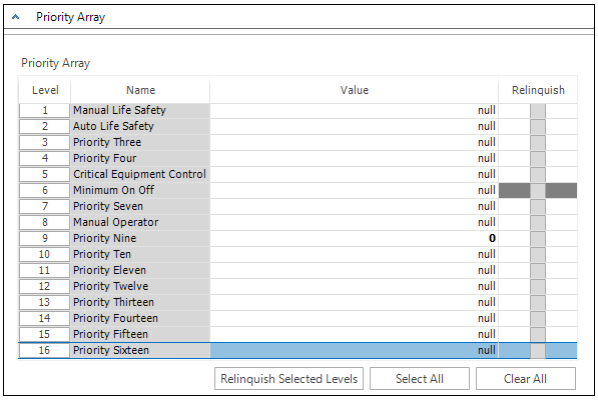
Priority Array Properties

The Priority Array property holds a priority list for writing values for the object.

To relinquish a priority value, select the check box and then click **Relinquish**.

For an explanation, see the topic *Priority arrays* on page 402.

Illustration 35–3 Priority Array Properties



Alarm Routing Properties

Alarm classes designate a group of operators who receive notifications of alarms and events originating from this object. See the topics *An overview of notifications, alarms, and events* on page 133 and *Emailing operator alarms* on page 171.

Output objects–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 401
- *Priority arrays* on page 402
- *Output objects-analog* on page 491
- *Output objects-binary* on page 494

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. Printable characters must be used for the Name property.

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 402.

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 requirements for this object. The definition of the profile is unique to the vendor 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 402.

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 objects* on page 490 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 35–19 Multistate output 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
- 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.

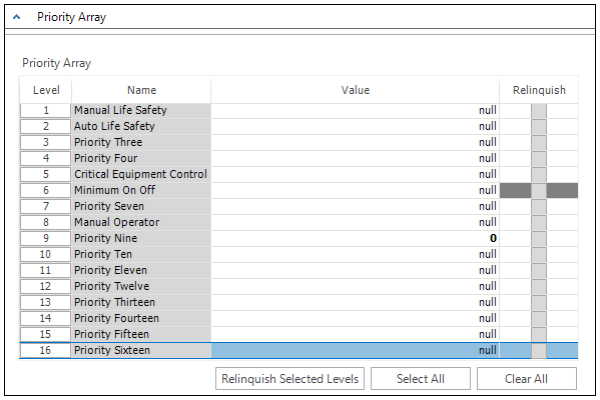
Priority Array Properties

The Priority Array property holds a priority list for writing values for the object.

To relinquish a priority value, select the check box and then click **Relinquish**.

For an explanation, see the topic *Priority arrays* on page 402.

Illustration 35–4 Priority Array Properties



Alarm Routing Properties

Alarm classes designate a group of operators who receive notifications of alarms and events originating from this object. See the topics *An overview of notifications, alarms, and events* on page 133 and *Emailing operator alarms* on page 171.

Program objects

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 217
- *Programming with the Code Editor* on page 195

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. Printable characters must be used for the Name property.

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 requirements for this object. The definition of the profile is unique to the vendor 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 35–20 Control Basic program states

State	Description of state
IDLE	The program is not executing
LOADING	The program is being loaded.
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 Program Change 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 35–21 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 35–22 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 35–23 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 Overridden 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 objects

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. Printable characters must be used for the Name property.

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 506 or one of the *Exception Schedules*.

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 requirements for this object. The definition of the profile is unique to the vendor 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 402.

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 35–24 Schedule 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 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 Controls 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 objects* on page 441.

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 506.

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. Printable characters must be used for the Name property.

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 requirements for this object. The definition of the profile is unique to the vendor 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 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 objects

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 109
- *Trend Log Multiple objects* on page 514

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. Printable characters must be used for the Name property.

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 requirements for this object. The definition of the profile is unique to the vendor 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 a second — 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 objects* on page 490 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 35–25 Trend Log 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	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 contains 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

Alarm classes designate a group of operators who receive notifications of alarms and events originating from this object. See the topics *An overview of notifications, alarms, and events* on page 133 and *Emailing operator alarms* on page 171.

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 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 has been collected.

Export To export log data as comma separated values, click **Export**. Each data value is exported with a timestamp.

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 109
- *Trend Log objects* on page 509

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. Printable characters must be used for the Name property.

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 requirements for this object. The definition of the profile is unique to the vendor 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 objects* on page 490 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 35–26 Trend Log Multiple 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	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

Alarm classes designate a group of operators who receive notifications of alarms and events originating from this object. See the topics *An overview of notifications, alarms, and events* on page 133 and *Emailing operator alarms* on page 171.

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.

Export To export log data as comma separated values, click **Export**. Each set of data values are exported with a time stamp.

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 401
- *Value objects-binary* on page 521
- *Value objects-multistate* on page 525

General Properties

Object Instance The value object number. Value 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. Printable characters must be used for the Name property.

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 402.

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 requirements for this object. The definition of the profile is unique to the vendor 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 402.

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 565.

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 objects* on page 490 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.

Analog value 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. 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.

Event Time Stamps This property holds the times of the last event notifications of *TO OFF NORMAL, TO FAULT,* and *TO NORMAL* events.

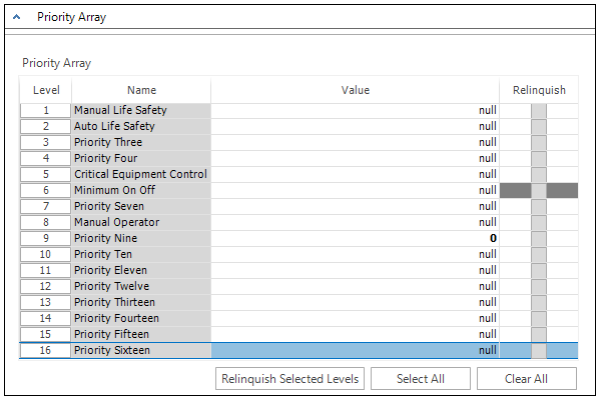
Priority Array Properties

The Priority Array property holds a priority list for writing values for the object.

To relinquish a priority value, select the check box and then click **Relinquish**.

For an explanation, see the topic *Priority arrays* on page 402.

Illustration 35–5 Priority Array Properties



Alarm Routing Properties

Alarm classes designate a group of operators who receive notifications of alarms and events originating from this object. See the topics *An overview of notifications, alarms, and events* on page 133 and *Emailing operator alarms* on page 171.

Value objects–binary

A binary value object is a standard BACnet object whose properties represent a binary value residing in memory. Use value objects to define program variables in Control Basic. Variables are placeholders for information such as flags and conditions.

Related topics

- *BACnet objects and properties* on page 401
- *Value objects-analog* on page 518

- *Value objects-multistate* on page 525

General Properties

Object Instance The value object number. Value 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. Printable characters must be used for the Name property.

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 402.

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 requirements for this object. The definition of the profile is unique to the vendor 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 402.

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 objects* on page 490 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 35–27 Binary value 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. 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.

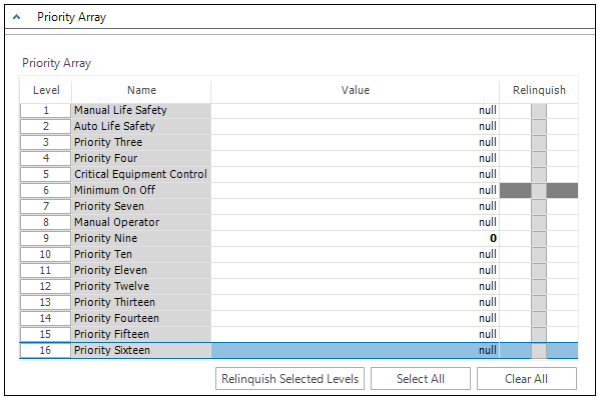
Priority Array Properties

The Priority Array property holds a priority list for writing values for the object.

To relinquish a priority value, select the check box and then click **Relinquish**.

For an explanation, see the topic *Priority arrays* on page 402.

Illustration 35–6 Priority Array Properties



Alarm Routing Properties

Alarm classes designate a group of operators who receive notifications of alarms and events originating from this object. See the topics *An overview of notifications, alarms, and events* on page 133 and *Emailing operator alarms* on page 171.

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 401
- *Value objects-analog* on page 518
- *Value objects-binary* on page 521

General Properties

Object Instance The Multistate Value object number. Multistate Value 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. Printable characters must be used for the Name property.

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 402.

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 requirements for this

object. The definition of the profile is unique to the vendor 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 402.

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.



Note: KMC Conquest Controllers support up to 16 states.

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 objects* on page 490 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 represents the time, in seconds, that the Off Normal conditions must exist before an Off Normal event state is indicated.

Time Delay Normal Time Delay Normal represents the time, in seconds, that the Normal conditions must exist before a Normal event state is indicated.

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 35–28 Multistate Value 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. 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 the Event State property 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.

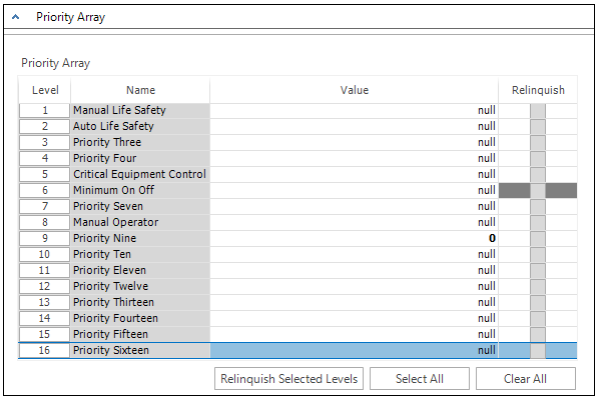
Priority Array Properties

The Priority Array property holds a priority list for writing values for the object.

To relinquish a priority value, select the check box and then click **Relinquish**.

For an explanation, see the topic *Priority arrays* on page 402.

Illustration 35–7 Priority Array Properties



Alarm Routing Properties

Alarm classes designate a group of operators who receive notifications of alarms and events originating from this object. See the topics *An overview of notifications, alarms, and events* on page 133 and *Emailing operator alarms* on page 171.

Section 36: Configuring Sensor Port objects

This section covers the configuration of NetSensors and the HPO-9007 wireless gateway.

Sensor Port Objects are active devices that plug directly into a KMC Controls BACnet controller. Use one of the following configuration procedures to configure Sensor Port Objects.

- **KMD-1100 and KMD-1200 series** NetSensors are used with the BAC-5800 and BAC-7000 series controllers. Configuration is described in the topic *Configuring KMD series NetSensors for BACnet* on page 529.
- **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 535.
- **Wireless gateway** *Configuring the HPO-9007 wireless gateway* on page 541.

See also the topic *Configuring a NetSensor for KMDigital controllers* on page 599.

Configuring KMD series NetSensors for BACnet

The KMD-1160, KMD-1180, KMD-1260, and KMD-1280 series NetSensors connect directly to any of the following KMC BACnet controllers:

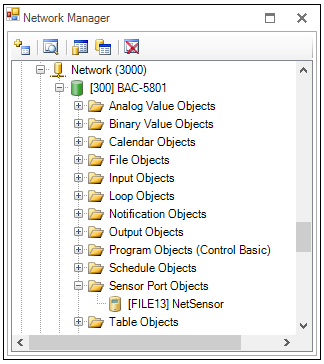
- BAC-5800 series
- BAC-7000 series
- BAC-7400 series

NetSensors are wall-mounted display and sensor units that consist of an LCD display, a thermistor, up to seven buttons, and optional humidity and motion sensors. Additional information for the NetSensor can be found in the following documents.

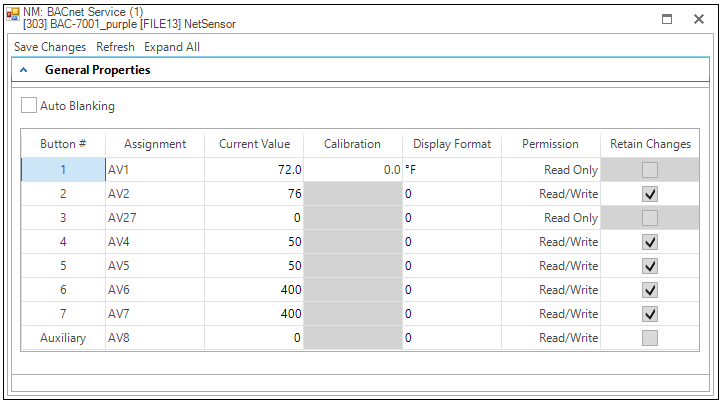
- The NetSensor installation guide.
- The Control Basic topic *NETSENSORSTATUS* on page 300.

To configure a NetSensor object, do the following:

1 In the Network Manager list, click ⊞ or ⊟ to expand or collapse the list to open the Sensor Port Objects folder.



2 Double-click the NetSensor icon to open the NetSensor object. The NetSensor tab opens.



Button #	Assignment	Current Value	Calibration	Display Format	Permission	Retain Changes
1	AV1	72.0	0.0	°F	Read Only	☐
2	AV2	76		0	Read/Write	☑
3	AV27	0		0	Read Only	☐
4	AV4	50		0	Read/Write	☑
5	AV5	50		0	Read/Write	☑
6	AV6	400		0	Read/Write	☑
7	AV7	400		0	Read/Write	☑
Auxiliary	AV8	0		0	Read/Write	☐

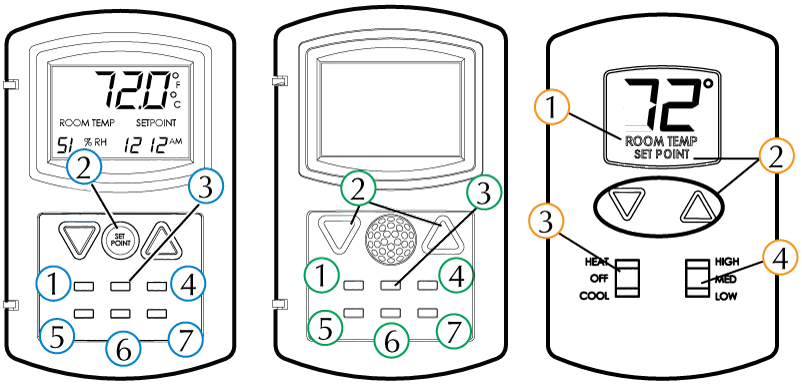
3 As required, configure each of the buttons. See *Button configuration examples* on page 530 for suggestions on configuration.

4 When finished, click **Save Changes**.

Button configuration examples

Configuring a KMD series NetSensor consists of assigning an object—typically a value object—to one of the buttons on the NetSensor. The value of the object is then displayed by the NetSensor and controlled with the assigned button. For a description of the NetSensor button functions, see *NetSensor button functions* on page 532.

Illustration 36–1 NetSensor button layouts



Button 1–internal temperature sensor Button 1 and the space temperature sensor inside of the NetSensor are usually associated with Analog Value object AV1.

1. In **Assignment**, select an Analog Value object to associate with the temperature button or drag an object from the Network Manager list to **Assignment**.
2. Set **Display Format** to **°F** or **°C**.
3. Set **Permission** to **Read Only**. Operators can then view the room temperature by touching Button 1 but cannot change the value.

   If input calibration is required, see *Calibration* on page 533.

Button 1–controller points To manage or display an object value from Button 1, configure as follows:

1. 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**.
2. Set **Display Format** to **0, 0.0** or **0.00**.
3. Set **Permission** to **Read Only** or **Read/Write**.
   - If set to **Read Only**, operators can view the value associated with Button 1 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.

1. In **Assignment**, select an Analog Value object to associate with the setpoint button or drag an object from the Network Manager list to **Assignment**.
2. Set **Display Format** to **0, 0.0** or **0.00**.
3. 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.
4. In a Program object, add 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.

1. In **Assignment**, select an Analog Value object to associate with the humidity button or drag an object from the Network Manager list to **Assignment**.
2. Set **Display Format** to **0.**
3. 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.

1. Set **Display Format** to **Time** which will automatically format the display with a colon (:).
2. In **Assignment**, select an Analog Value object to associate with the time button or drag an object from the Network Manager list to **Assignment**.
3. In a Program object, 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 300.

```
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.

1. Under **Assignment**, select an Analog Value object to associate with the motion sensor. Typically this is Analog Value object AV8.
2. Set **Display Format** to **0**.
3. Set **Permission** to **Read/Write**.
4. In a Program object, add 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 example detects motion and changes the state of Binary Value object 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
```

NetSensor button functions

The functions of the NetSensor buttons are listed in the following table.

Table 36–1 NetSensor button functions

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.

Table 36–1 NetSensor button functions (continued)

Button	Function
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.**

NetSensor tab properties

The properties in the NetSensor tab define how each of the buttons function.

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 (continued)* 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 36–2 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.

Table 36–2 NetSensor analog display formats (continued)

Unit	Action and display
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 36–3 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.

Configuring the STE-9000 series NetSensor

The STE-9000 series NetSensor is a wall-mounted display and sensor unit that connects directly to any of the following Conquest controllers.

- BAC-5900 series
- BAC-9000 series
- BAC-9300 series

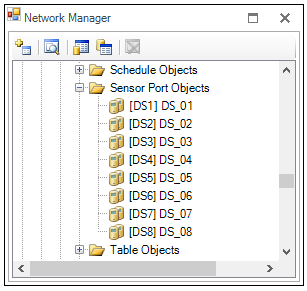
The sensor consists of an LCD display (on models without the NDL suffix), a thermistor, control buttons (on models without the NDL suffix), and optional humidity, motion sensor, and CO2 sensors. By installing an HPO-9001 NetSensor Distribution board, up to eight NetSensors can be configured for each controller. Additional information for the NetSensor can be found in the following documents.

- The installation guide for the NetSensor.
- The installation guide for the HPO-9001 distribution module.

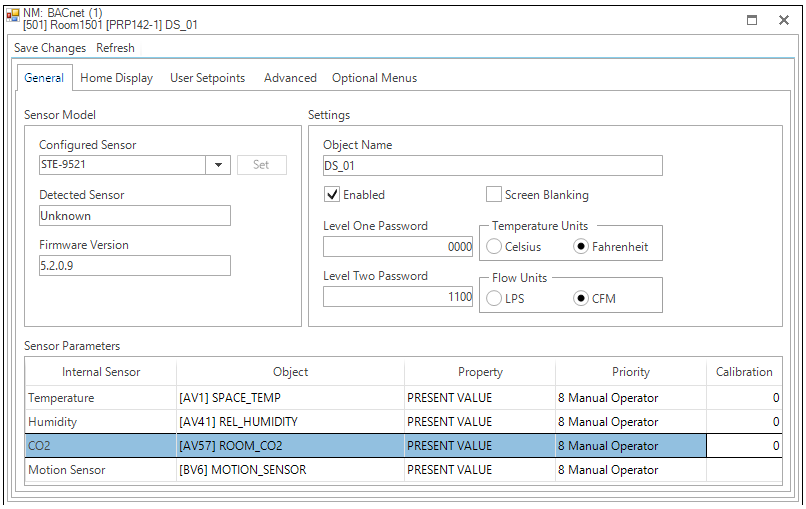
Caution: BAC-9000 and BAC-9300 series controllers are programmed for specific display functions with an STE-9000 sensor. Changing the sensor configuration may disrupt controller operation. Refer to the controller application guide before changing NetSensor configuration.

To configure a NetSensor, do the following:

1. In the Network Manager list, locate and open a device that includes a NetSensor.
2. Locate and open the Sensor Port Objects folder.



3 Double-click the NetSensor icon. The NetSensor configuration tab opens.



4 Make changes as needed.

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

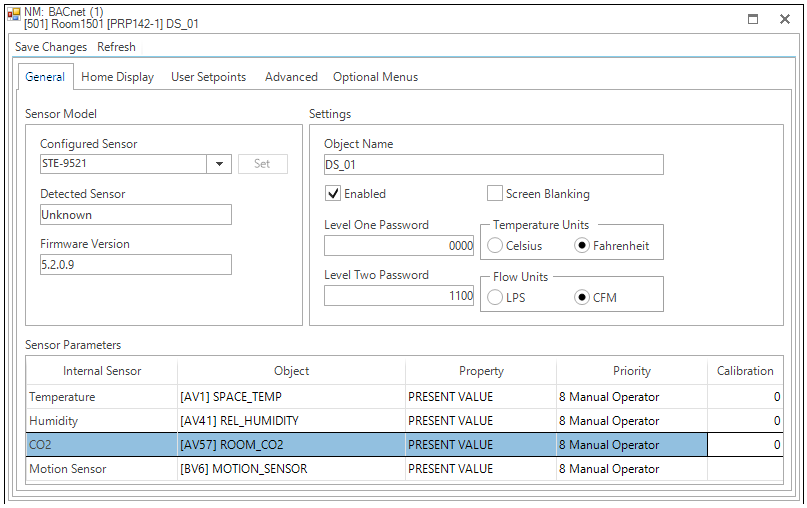
6 Right-click the device icon.

7 From the drop-down menu, choose **Reinitialize Device**.

General tab

The General tab displays information about the sensor and sets overall operating properties.

Illustration 36–2 Conquest NetSensor General tab



Sensor model Properties in this area list basic properties for the sensor.

Configured Sensor Selects the default configuration for each sensor model.

Detected Sensor Shows the model of the sensor connected to the controller.

Firmware Version Shows the firmware version.

Settings The properties in the Settings area set up security and basic display information.

Object Name Enter the name of the sensor 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.

Enabled By default, this check box is selected for NetSensor 1 in each controller. If additional NetSensors are connected with an HPO-9001 NetSensor distribution module, select the Enabled check box.

Temperature Units Select the temperature display units as either degrees Fahrenheit or Celsius.

Flow Units Select the flow units as either CFM (Cubic Feet per Minute) or LPS (Liters Per Second).

Sensor Parameters The properties in the Sensor Parameters list associate a sensor function with a value object in the controller.



Note: When the NetSensor is connected to a controller, the internal sensors for temperature, humidity, CO2, and motion are automatically associated with the default objects for Temperature, Humidity, CO2, and Motion Sensor. Changing these assignments will disrupt the sequence of the factory supplied programs.

Internal Sensor The type of physical internal sensor. Not all sensor types are included in every NetSensor.

Object and Property The object and property within the NetSensor to associate with an internal sensor.

Priority The level in the object's priority where the internal value will be stored.

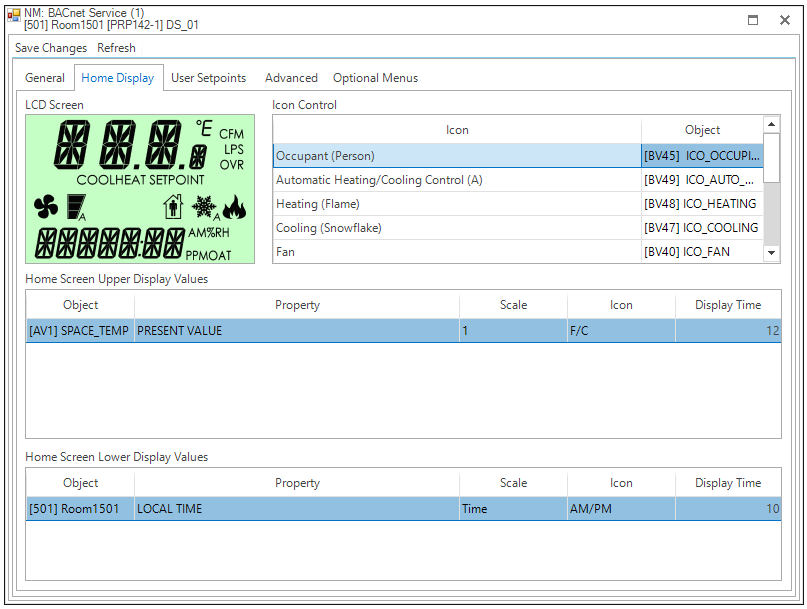
Calibration Enter a calibration value for the temperature sensor. The range is +/- 20 degrees.

Home Display tab

Items in the Home Display tab associate the properties in the controller to display on the various parts of the sensor LCD screen.

Note: The Home Display tab is not applicable to NDL model NetSensors.

Illustration 36–3 Conquest NetSensor Home Display tab



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.

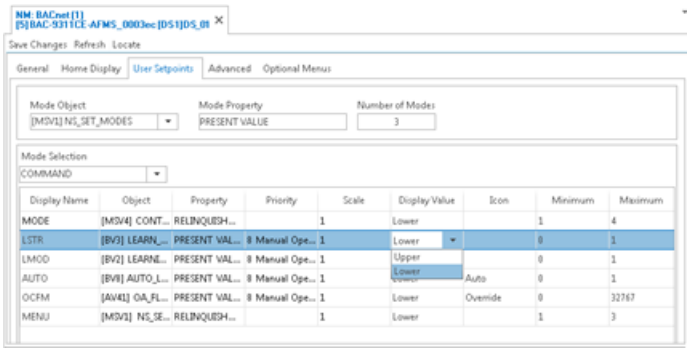
Home Screen Upper Display Values This is a list of objects whose Present Value properties are displayed in the upper display of the home screen. The time of display for each object is set by the value in the column **Display Time**.

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

The User Setpoints tab sets the scale, icons, and limits for the setpoints available with a Level 1 password.

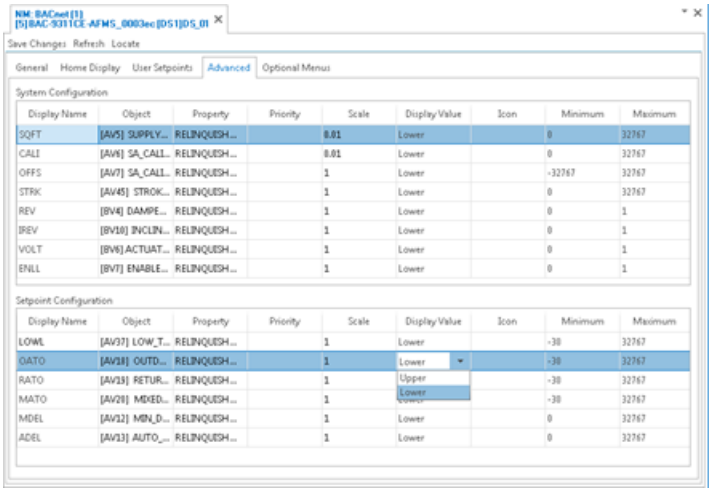
Illustration 36–4 Conquest NetSensor User Setpoints tab



Advanced tab

Items under the Advanced tab associate a sensor Configuration or Setpoint menu item with an object in the controller. See the documentation for the NetSensor and controller for the details on each menu item.

Illustration 36–5 Conquest NetSensor Advanced tab



System Configuration Items in this list associate a value object with a system configuration function such as fan or reheat type.

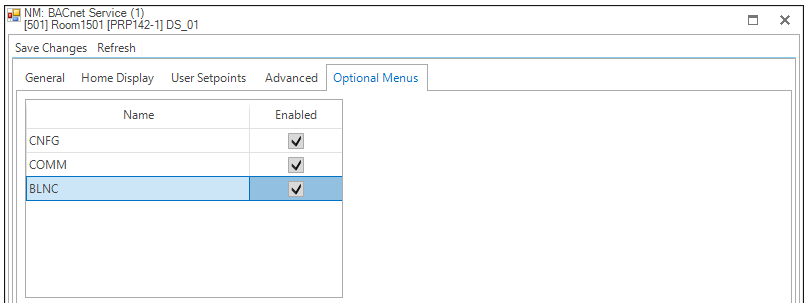
Setpoint Configuration This list designates the value objects and values for setpoint limits.

Optional Menus tab

The properties under the Optional Menus tab enables or disables the CNFG (Configuration), COMM (Communication), or BLNC (Balancing) menus in the sensor. When a menu is not enabled, users are prevented from changing properties from the sensor in the disabled menus. See the documentation supplied with the sensors and controllers for the properties that are included in each type of menu.

Note: The BLNC (Balancing) menu is available only for VAV controllers.

Illustration 36–6 Conquest NetSensor Optional Menus tab



Configuring the HPO-9007 wireless gateway

The HPO-9007 gateway is a receiver for STW-6000 and THW-6000 series of wireless sensors. It is compatible also with EnOcean sensors from other manufacturers. Connect the gateway directly to a Room Sensor port or to an HPO-9001 distribution module. The HPO-9007 is compatible with any of the following Conquest controllers.

- BAC-5900 series
- BAC-9000 series
- BAC-9300 series

Before you begin, review the information provided with the HPO-9007 gateway, the sensors, and, if used, the HPO-9001 distribution module.

You will also need the following items and information:

- Access to the Learn button inside of the sensors.
- The sensor identification number located inside of the sensors.
- An unassigned Analog Value object for each sensor parameter to be used by the controller or site.

Configuring sensors is a two-part process:

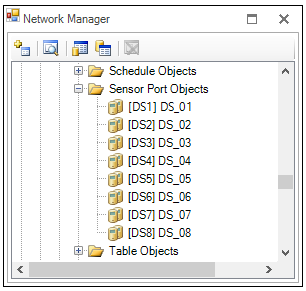
1. Converting a sensor object in the Sensor Port Object list to a gateway object. This is only done once for each gateway.
2. Adding individual sensors to the gateway.

Converting a sensor object

A NetSensor object in the Sensor Port Objects folder must be converted to a gateway object before sensors can be added to it.

To convert a sensor object to a gateway object, do the following:

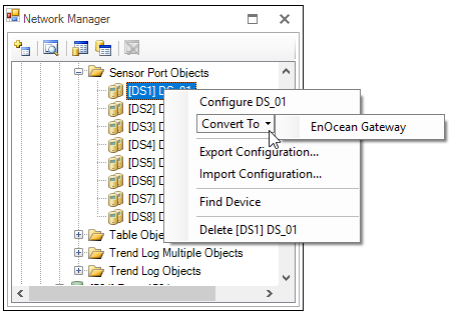
1 In the Network Manager list, click ⊞ or ⊟ to expand or collapse the network list to locate the Sensor Port Objects folder.



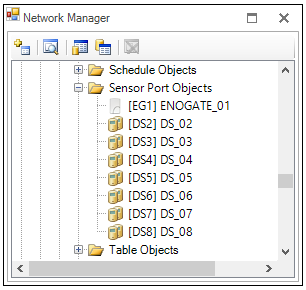
2 Right-click a NetSensor icon .

3 From the drop-down menu, choose **Convert To**.

4 Click **EnOcean Gateway**.



The NetSensor icon changes to a gateway icon .

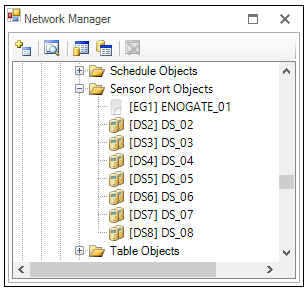


5 Once a gateway is in the Network Manager list, add compatible sensors to it.

Adding sensors by discovery

1 In the Network Manager list, click ⊞ or ⊟ to expand or collapse the network list to locate the Sensor Port Objects folder.

2 Locate the gateway icon in the Sensor Port Objects list. If the gateway object is not in this list, see *Converting a sensor object* on page 542.

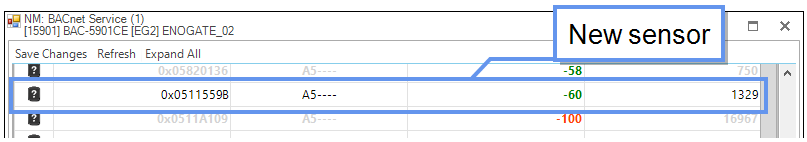


3 Double-click the icon to open the gateway configuration tab.

4 Expand the **Gateway Sensor List Properties**.

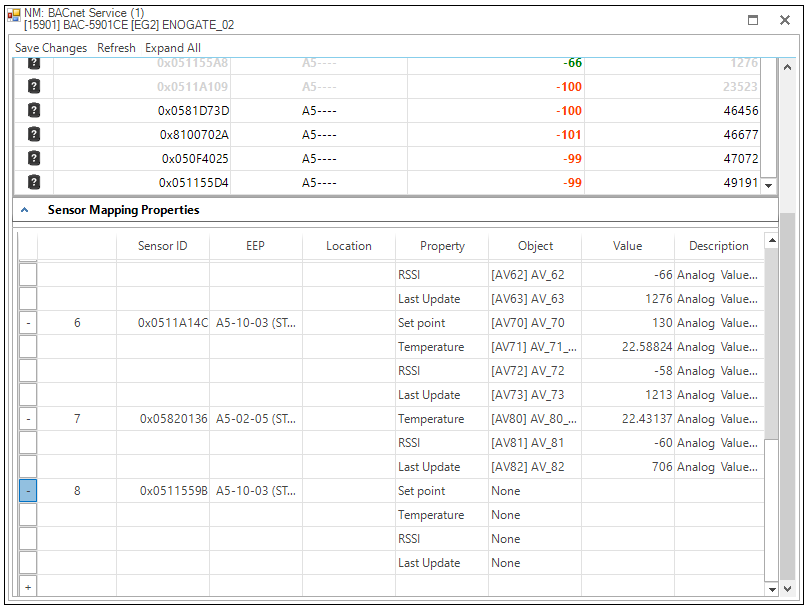
5 Inside of a sensor, press and release the **Learn** button.

When the gateway has detected the sensor, the sensor ID and other data will be added to the Gateway Sensor List in normal type.

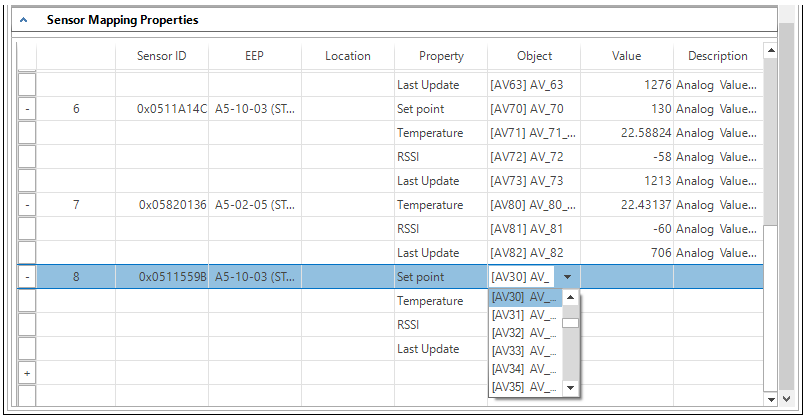


6 Expand the **Sensor Mapping Properties** area.

7 From the Gateway Sensor List area, drag the sensor icon to any area of the Sensor Mapping Properties area. Once the sensor is in the Sensor Mapping properties area, the sensor text in the Gateway Sensor List Properties turns gray.



8 From the text list in the Object column, choose an Analog Value object to associate with each sensor property. At least one property must be paired with an Analog Value object.



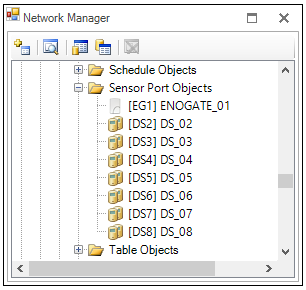
9 In **Location**, enter the physical location of the sensor.

10 Click **Save Changes** at the top of the tab.

Adding sensors by Sensor ID

Use the manual method to add sensors that may not be available or are currently out of range of the gateway. This includes adding compatible EnOcean devices from other manufacturers.

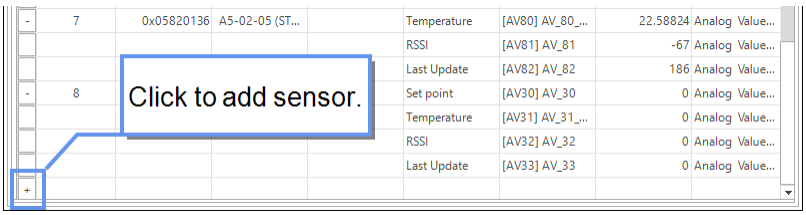
1 Locate the gateway icon in the Sensor Port Objects list. If the gateway object is not in the Sensor Port Objects list, see *Converting a sensor object* on page 542.



2 Double-click the icon to open the object.

3 Expand **Sensor Mapping Properties**.

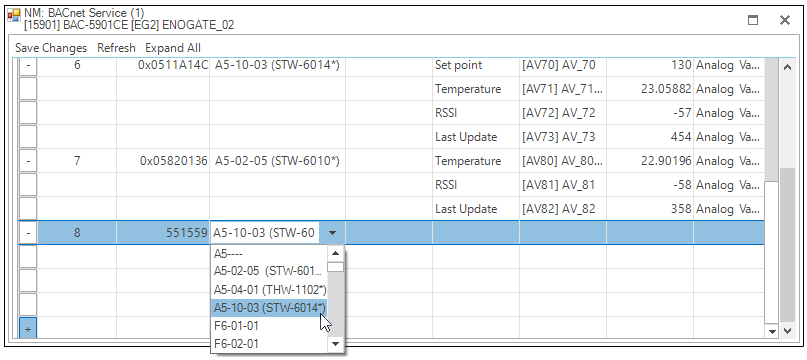
4 At the bottom of the list, click the plus button to add a new sensor.



5 In **SensorID**, enter the numerical portion of the sensor identification number.

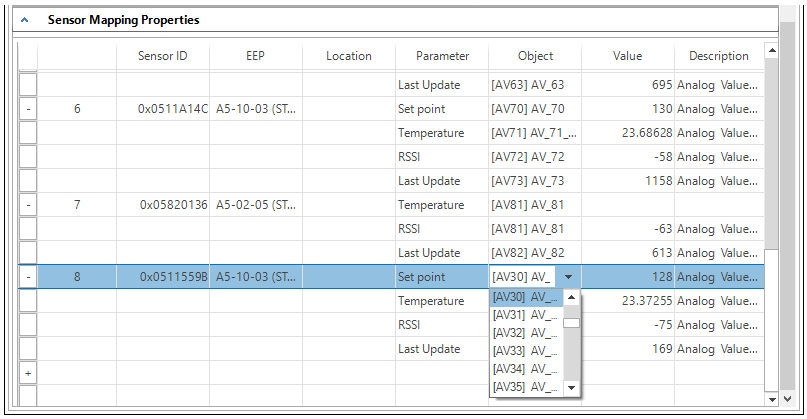
6 From the text list in the **EEP** column, choose the EnOcean Equipment Profile for the sensor.

- Models from KMC Controls, Inc. are listed by their model number.
- For other EnOcean sensors, choose the EEP provided by the manufacturer.



7 In **Location**, enter the physical location of the sensor.

8 From the **Object** column text list, choose an Analog Value object to associate with each sensor parameter. At least one parameter must be paired with an Analog Value object.

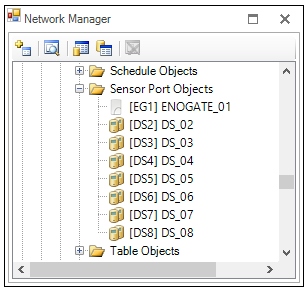


9 Click **Save Changes** at the top of the tab.

Adding sensors by using Add Sensor to Gateway

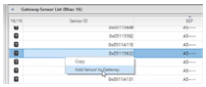
Use the Add Sensor to Gateway method to add sensors when computer display size is limited.

1 Locate the gateway icon in the Sensor Port Objects list. If the gateway object is not in the Sensor Port Objects list, see *Converting a sensor object* on page 542.



2 Double-click the icon to open the object.

3 Expand **Sensor Mapping Properties**.



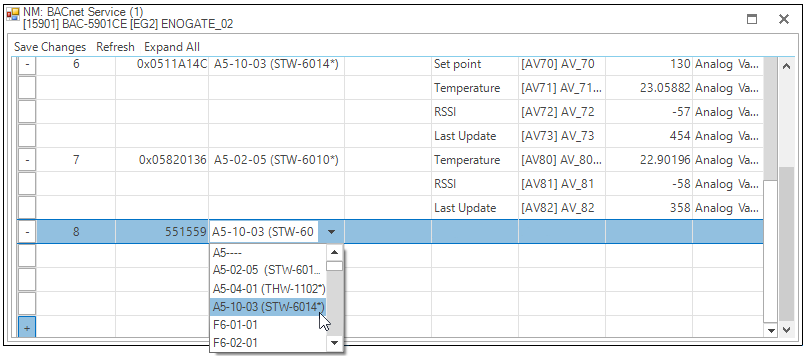
4 In the Gateway Sensor List, click to select the sensor to add.

5 Right-click anywhere in the selected row.

6 From the drop-down list, select **Add Sensor to Gateway**. The sensor appears in the Sensor Mappng Properties list.
Once the sensor is in the Sensor Mapping Properties area, the sensor text in the Gateway Sensor List Properties turns gray.

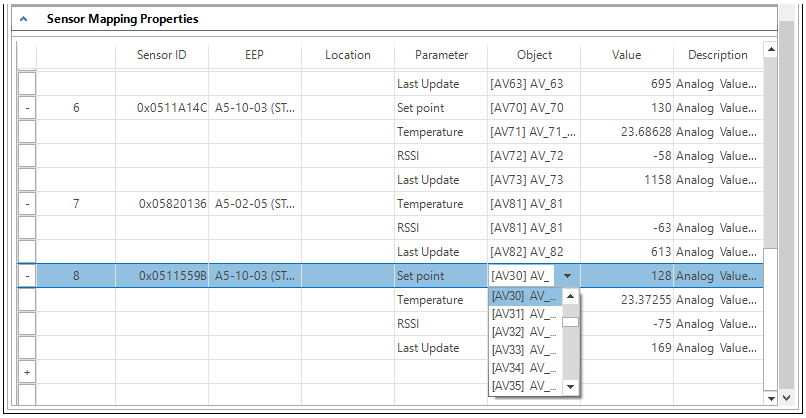
7 From the text list in the **EEP** column, choose the EnOcean Equipment Profile for the sensor.

- Models from KMC Controls, Inc. are listed by their model number.
- For other EnOcean sensors, choose the EEP provided by the manufacturer.



8 In **Location**, enter the physical location of the sensor.

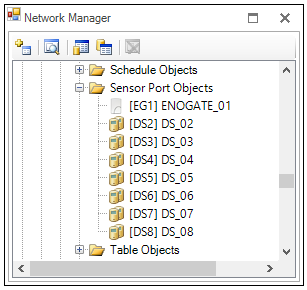
9 From the **Object** column text list, choose an Analog Value object to associate with each sensor parameter. At least one parameter must be paired with an Analog Value object.



10 Click **Save Changes** at the top of the tab.

Deleting sensors

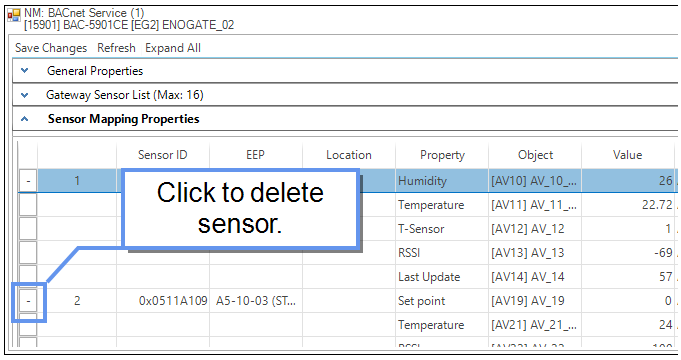
1 Locate the gateway icon in the Sensor Port Objects list.



2 Double-click the icon to open the object.

3 Expand **Sensor Mapping Properties**.

4 Click the minus button next to the sensor. The sensor will be deleted from Sensor Mapping Properties but will remain in Gateway Sensor List.



5 Click **Save Changes** at the top of the tab.

Control Basic conversion examples

To convert sensor temperature to degrees Fahrenheit or dial position data to a setpoint offset, use the following examples.

Celsius to Fahrenheit Conquest wireless sensors send temperature data to a gateway only in degrees Celsius. In the following example, Analog Value object AV6 holds the sensor temperature data in degrees Celsius. The calculation converts the value in AV1 to degrees Fahrenheit and stores the result in AV1.

```
REM AV6 Degrees C from sensor
AV1 = AV6 * 1.8 + 32
```

Dial position The dial position for the Conquest sensor is stored as a value that ranges from 0-255. This value is proportional to a dial position that ranges from 0-270 degrees. In the following example, Analog Value object AV5 holds the sensor temperature offset span in degrees. For example, enter 4 for an offset of ± 2 degrees. The calculation converts the position stored in AV3 to an offset that is stored in AV2.

```
REM AV5 = The offset
REM AV3 = The sensor dial value (0 thru 255)
REM AV2 = Resulting offset
AV2 = AV5 / (255 / AV3) - AV5 / 2
```

General Properties

The General Properties display and configure the gateway as an object within the BACnet controller to which it is connected.

Object Instance The number of the object. Gateway objects are numbered sequentially within a BACnet device.

Object Name A text label of 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 requirements for this object. The definition of the profile is unique to the vendor and not part of the BACnet standard.

Description A text description of the object. The set of characters entered for the Description property must be printable characters.

Enable By default, this check box is selected for the first gateway. If additional gateways are connected with an HPO-9001 distribution module, select the Enable check box for each additional gateway.

Detected Model The model name property is assigned by KMC Controls, Inc. to represent the model of the gateway.

Status Displays "1" when a gateway is connected and "0" when the gateway is disconnected.

Firmware Version Displays the firmware version number in the gateway. Check the KMC Controls website for the current version of firmware. When calling for technical support, have the firmware release number available.

Hardware ID This property identifies the gateway as the connected device.

Board Revision The revision level of the circuit board in the gateway.

Gateway Sensor List properties

The Gateway Sensor List displays information about any sensor that is within range of the gateway. Only 16 devices are displayed. To add a device not in the list, see *Adding sensors by Sensor ID* on page 544.

Sensor ID This is a unique identification number embedded in the sensor.

EEP The EnOcean Equipment Profile (EEP) identifies the device type and function of the sensor. The EEP is assigned by the manufacturer of the device and is part of each radio transmission from the sensor.

Signal (dB) The strength of the received signal. Signal is represented in a negative form; the closer the value is to 0, the stronger the received signal. The signal strength is color coded as follows:

Table 36–4 Signal strength

Color	Status	Signal strength
Green	Good	0 db to -75 dB
Yellow-orange	Fair	-76 db to -84 dB
Red	Poor	-85 dB or weaker

Seconds Since Rx This displays the number of seconds since the gateway received a signal from the sensor. Sensors from KMC Controls transmit when any of the following events take place:

- When temperature has changed by more than ±0.9° F (±0.5°C).
- When humidity changes ±2.0%.
- When the dial changes by ±10.5° of the 270° dial rotation.
- When the Learn button in the sensor is pushed.
- After 15-25 minutes of inactivity.

For more detailed information about when sensors transmit data, see the instructions supplied with the sensors.

Sensor Mapping Properties

The Sensor Mapping Properties panel displays information about a sensor. It is used also to associate a sensor property with an Analog Value object.

Sensor ID This is a unique identification number embedded in the sensor.

EEP The EnOcean Equipment Profile (EEP) identifies the device type and function of the sensor. The EEP is assigned by the manufacturer of the device and is part of each radio transmission from the sensor.

Location The physical location of the sensor. This is entered during configuration.

Property The Property is the raw data transmitted by the sensor. Sensors from KMC Controls, Inc. will include one or more of the following properties.

Set Point In sensors from KMC Controls, this is the position of the dial. Set Point ranges from a value of 0-255 which is proportional to 0-270° of rotation. See *Control Basic conversion examples* on page 547.

Temperature The space temperature in Celsius. See *Control Basic conversion examples* on page 547.

Humidity The relative humidity, in percent, of the space.

RSSI Signal strength in dB and corresponding to *Signal (dB)* in the *Gateway Sensor List* area.

Last Update The number of seconds since the gateway received a signal from the sensor.

Object The Object text list pairs the Property with an Analog Value object. When new sensor data is received, the Analog Value object is updated at Priority 8.

Value The most recent value received from the sensor.

Description This is the Description property of the Analog Value object that is paired with the sensor Property. This can only be changed by editing the Analog Value object. See *Configuring BACnet devices and objects* on page 410.

Section 37: Configuring Conquest FlexStats

This section covers the configuration of Conquest FlexStats using the Display object.

The BAC-19xxxx series Conquest FlextStat is a wall-mounted touchscreen display and sensor unit that connects directly to any of the following Conquest controllers.

- BAC-5900(A) series
- BAC-9000(A) series
- BAC-9300(A) series

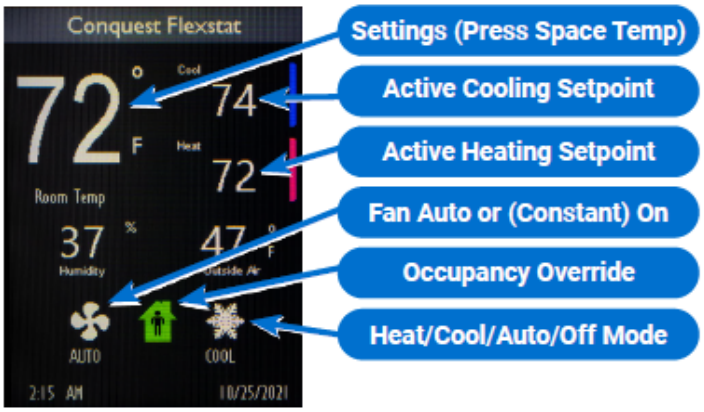
The sensor consists of a resistive touchscreen color display, a thermistor, and optional humidity, motion, and CO2 sensors.

Onscreen menus allow basic display customization for features such as color scheme, brightness, and contrast. KMC Connect software provides more extensive customization options using the Display object, a proprietary BACnet object.

Additional display features that can be customized through software are:

- a title bar
- room temperature and setpoint
- a four-object icon button bar
- time and date
- up to 24 total rotation value displays (e.g., humidity, C02, discharge air temperature) in three areas of eight current values each.

Illustration 37–1 Conquest FlexStat – default screen configuration shown



To configure a Conquest FlexStat, do the following:

1. In the Network Manager list, locate and open the BAC-19xxxx.
2. Click the plus icon ⊞ to the left of the Display Object to reveal the Display Objects folder.

3 Double-click the UI (User Interface) icon.

4 Make changes as needed. Refer to *Conquest FlexStat Configuration Tabs* on page 552.

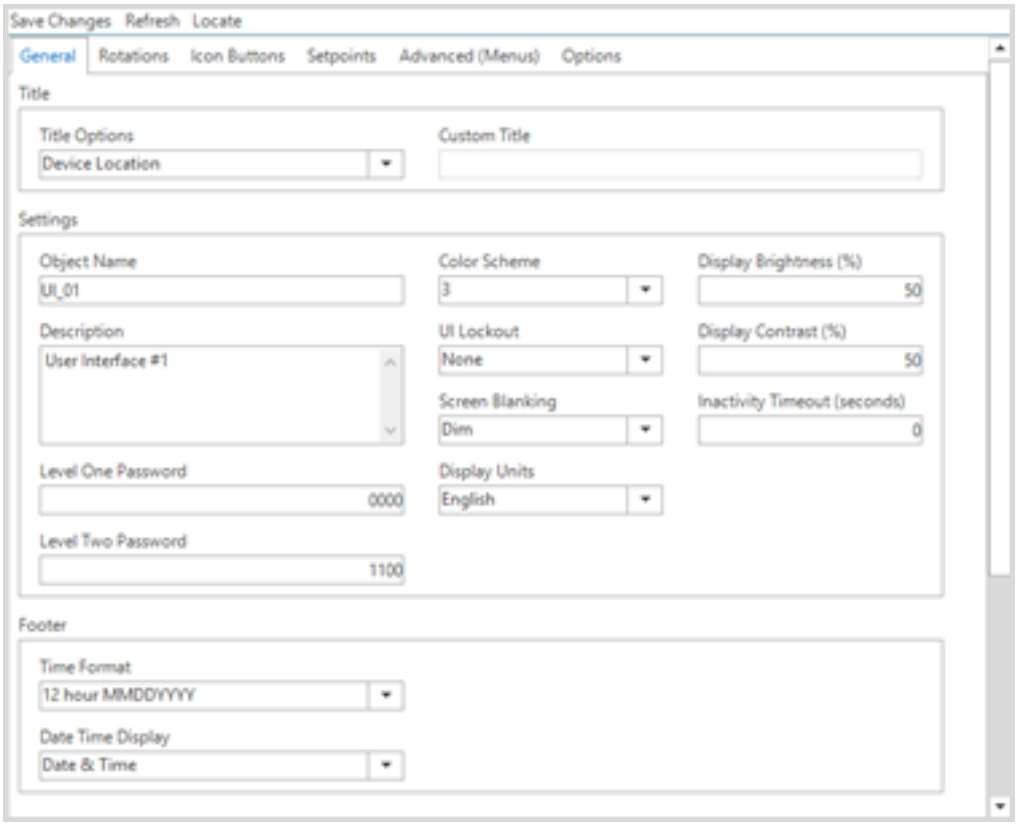
5 When changes are complete, click **Save Changes** near the top left corner of the page.

6 In the Network Manager, right-click the device icon for the BAC-19xxxx.

7 From the shortcut menu, choose **Reinitialize Device**.

Conquest FlexStat Configuration Tabs

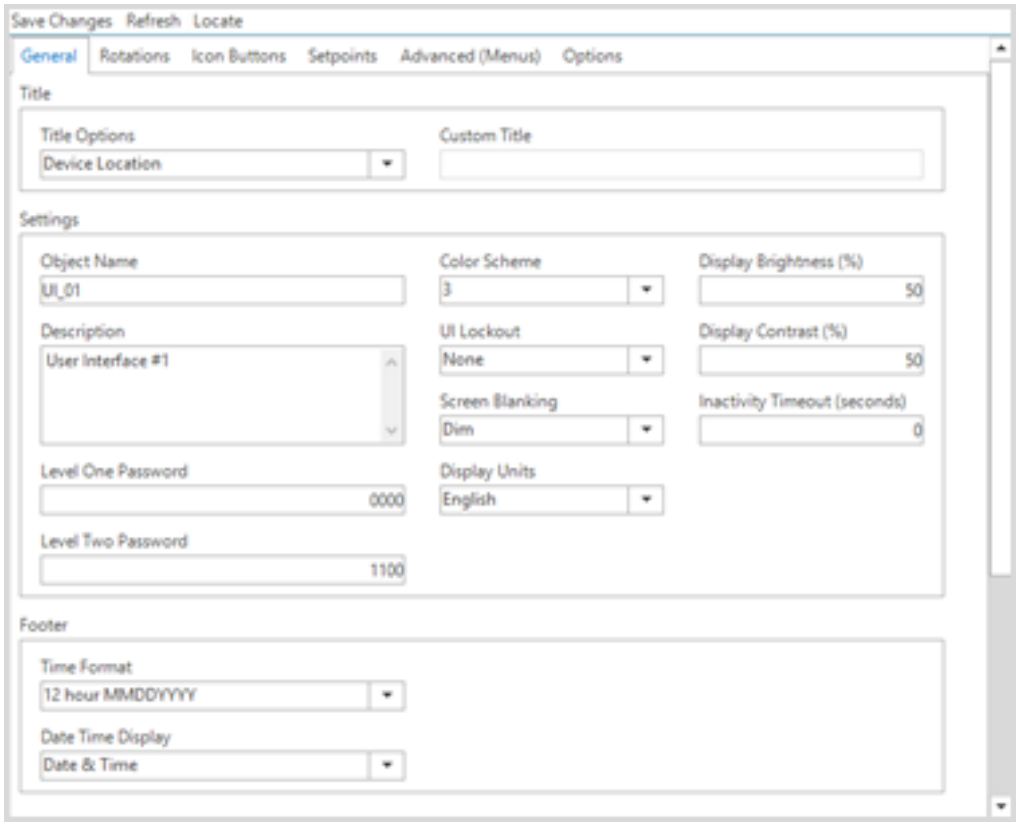
This topic explains the tabs in the Conquest FlexStat Display Object configuration page.



General tab

The Display Object configuration page opens to the General tab. The General tab displays information about the Conquest FlexStat and sets overall operating properties.

Illustration 37–2 Conquest FlexStat General tab



Title Properties in this area determine how the title appears at the top of the display.

Title Options Use the drop-down arrow to open and then select from a list of the following options.

Device Location Identifies the device by physical location (e.g., boiler room, chiller plant, etc.). Selecting this option populates the title with the property name as it is listed in the Network Manager.

Device Name Identifies the device model name (e.g., BAC-5901AC)

Device Description Identifies the device by function (e.g., East Wing VAV)

Custom Title Type a custom title in this box.

Settings Properties in this area set up basic display information and security features.

Object Name Enter the name of the sensor to be shown in the Network Manager list.

Description Enter a description of the display object.

Level One Password Enter the numerical Password One for the user-accessible setpoints. This should be four digits.

Level Two Password Enter the numerical Password Two for the configuration functions used by a controls technician. This should be four digits.

> **Note:** In KMC Connect, both security levels can be viewed and changed without having to enter either password first.

Color Scheme Sets the color scheme for the display panel.

UI Lockout When enabled, prevents access from the display panel to either the Icon Buttons or Icon Buttons and Setpoints.

Screen Blanking When enabled, the screen darkens automatically after a selected number of seconds. The options below determine what is applied after an Inactivity Timeout.

None The screen remains illuminated.

Dim The screen dims to a predefined level set in firmware.

Backlight Off The backlight is off. To re-enable the backlight, press anywhere on the display.

Inactive Screen Only the space temperature is illuminated on the display.

Display Units Select units of measurement as either English or Metric.

Display Brightness Sets the display brightness as a percentage.

Display Contrast Sets the display contrast as a percentage.

Inactivity Timeout Sets the time in seconds since the last user contact for Screen Blanking to apply.

Footer Sets how the time and date display at the foot of the screen.

Time Format Sets the format for time shown in the lower left corner of the display. Options are 12 Hour (MM/DD/YYYY), 12 Hour (DD/MM/YYYY), 24 Hour (MM/DD/YYYY), or 24 Hour (DD/MM/YYYY) format.

Date Time Display Sets the date and time format. Options are Date & Time, Time Only, Date Only, or None.

> **Note:** When a Conquest FlexStat is unpowered long enough to lose the RTC (Real Time Clock) time, it will revert to the default time and date of 12:00 AM 01/01/2000.

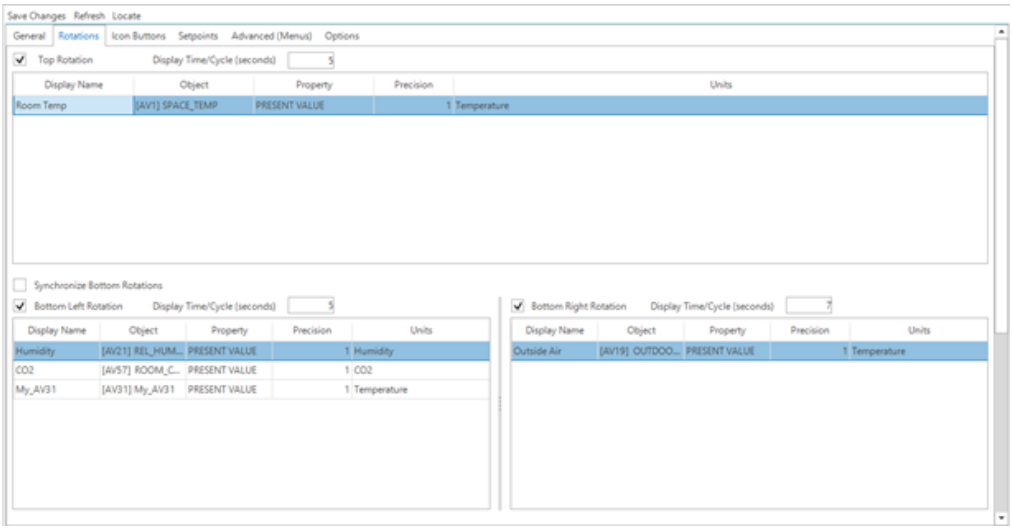
If the time and/or date is incorrect, do any of the following.

- Manually set the time and date from the display menu.
- Check the DST (Daylight Saving Time) settings.
- Using KMC software, send a time synchronization command. Refer to *Setting BACnet system time* on page 417.

Rotations tab

The Rotations tab lists the settings for each of the three rotation areas in the display (Top, Bottom Left and Bottom Right). To add an object to be displayed in rotation, drag it from the Network Manager to the appropriate panel.

Illustration 37–3 Conquest FlexStat Rotations tab



Top Rotation When selected, this checkbox activates the rotation settings for the Top portion of the display.

Bottom Left Rotation When selected, this checkbox activates the rotation settings for the Bottom Left portion of the display.

Bottom Right Rotation When selected, this checkbox activates the rotation settings for the Bottom Right portion of the display.

Display Time/Cycle (seconds) Sets the time in seconds for each feature to display in rotation.

Display Name Shows the name of the object added.

Property Shows the property to be displayed (e.g., PRESENT VALUE).

Precision Sets the precision of the numbers displayed to either one or two decimal places.

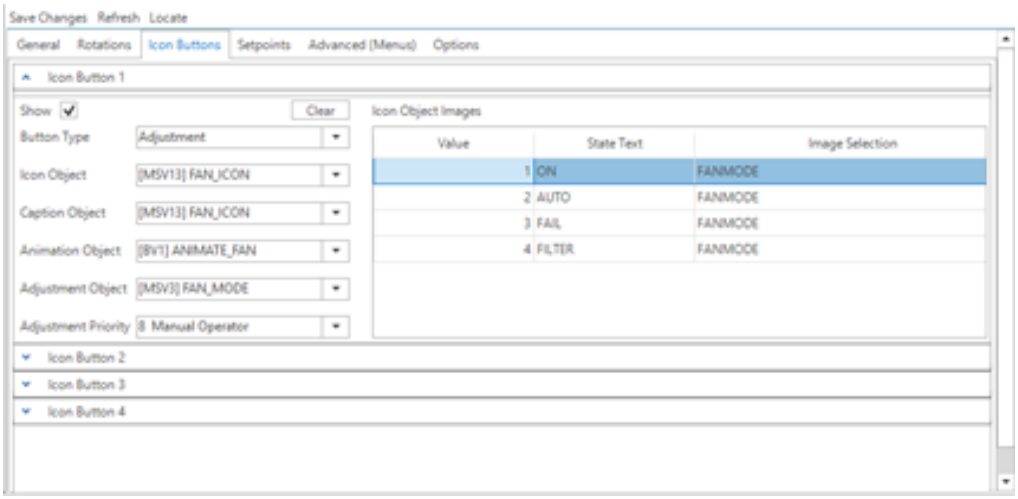
Units Shows the units of measurement to be applied to the object value (e.g., Temperature, Humidity, CO2).

Synchronize Bottom Rotations Selecting this checkbox synchronizes the rotation of the Bottom Left and Bottom Right rotation areas of the display. When the checkbox is selected, only the Display Time/Cycle of the bottom left rotation area is used. When unselected, each of the lower display areas updates according to its individual Display Time/Cycle.

Icon Buttons tab

The Icon Buttons tab is used to adjust the settings of each of the four customizable icon buttons.

Illustration 37–4 Conquest FlexStat Icon Buttons tab



Show When selected, shows the icon button on the display screen. When unselected, hides the icon button on the display screen.

Clear Clears all properties in the Icon Buttons tab.

Button Type Opens a drop-down list from which to select the object used to control the icon images. The default is Adjustment.

Adjustment Sets the object to be adjusted by pressing the icon on the display screen. Options are:

- Fan
- Occupancy
- Heat/Cool
- Not shown

Special Sets the temperature to display in degrees Fahrenheit or degrees Celsius.

Icon Object Selects the object type available from the drop-down list.

Caption Object Selects the type of caption available from the drop-down list.

Animation Object Selects whether the icon will be a static or animated image.

Adjustment Object Selects the object to be acted upon when the user touches the display screen.

Adjustment Priority Selects the priority level at which to write the value selected from the display screen.

Icon Object Images A table that lists the image for the present value state of the object defined in the Object Icon field that will be shown on the NetSensor display. The factory default is three images. The Conquest NetSensor can hold up to four icons and their settings in memory.

To edit the value within a cell, click and type the desired numeral.

Right-click within a cell to open a drop-down menu with the following options.

- Copy
- Paste
- Edit
- Clear Value

Value A read-only field that displays the value enumeration of the object.

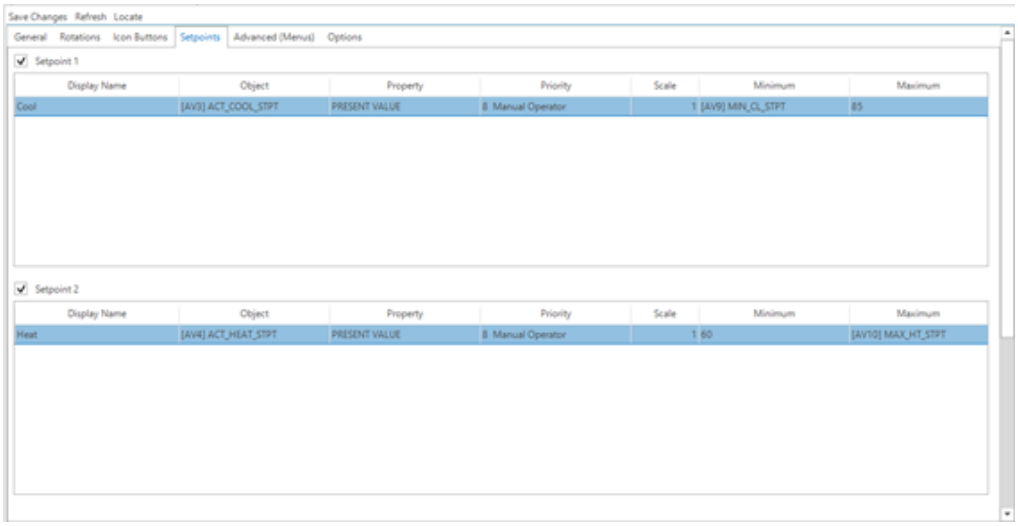
State Text A read-only field that displays the value of the state text for the given enumeration of the object.

Image Selection Displays the image that is selected for the given enumeration. This image displays on the touchscreen when the present value of the object matches the enumeration value. Clicking within the cell opens a drop-down list of available images.

Setpoints tab

The Setpoints tab sets the scale, icons and setpoint limits available with a Level 1 password.

Illustration 37–5 Conquest Flexstat Setpoints tab



Note: The space temperature sensor inside of the Conquest FlexStat is usually associated with Analog Value Object 1 (AV1).

There can be multiple setpoints accessible to the user. The default is two setpoints (e.g., heating setpoint and cooling setpoint).

Setpoint 1 When selected, allows touchscreen access to the first setpoint.

Setpoint 2 When selected, allows touchscreen access to the second setpoint.

Tip: One setpoint can be disabled if the FlexStat is used for cooling only or for only heating only.

Each setpoint has the following settings.

Display Name Shows the name of the object as it will appear on the FlexStat display.

Object Shows the name of the object associated with the setpoint.

Property Shows the name of the property associated with the setpoint.

Priority Shows the setpoint priority.

Scale Shows the scale associated with the setpoint. Multiplying the present value by the scale value yields a value in the selected units.

Minimum Shows the minimum value for the setpoint.

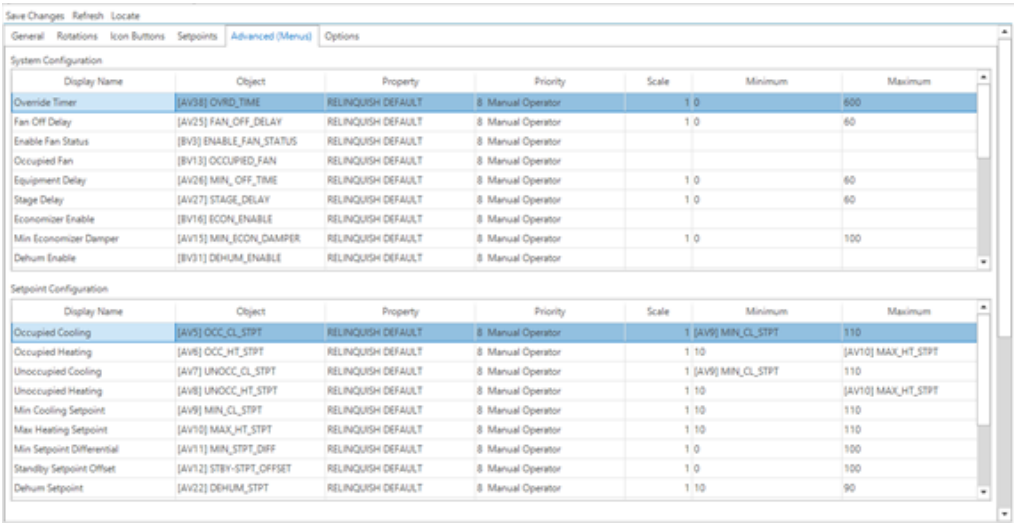
Maximum Shows the maximum value for the setpoint.

Advanced (Menus) tab

The properties under the Advanced (Menus) tab enable or disable the CNFG (Configuration), COMM (Communication), or BLNC (Balancing) menus in the FlexStat's display. Disabling a menu prevents users from changing its properties directly from the touchscreen.

Refer to the documentation supplied with the sensors and controllers for the properties that are included in each type of menu.

Illustration 37–6 Conquest FlexStat Advanced (Menus) tab



System Configuration Items in this list associate a value object with a system configuration function such as fan or reheat type. The exact parameters shown depend upon the application.

Setpoint Configuration This list designates the value objects and values for setpoint limits.

Both configuration types have the following parameters.

Display Name Shows the name of the object as it will appear on the FlexStat's display.

Object Shows the object name (e.g., [AV9] MIN_CL_STPT).

Property Shows the property name (e.g., RELINQUISH DEFAULT).

Priority Shows the object's priority level (e.g., 8 Manual Operator).

Scale Shows the scale for the object. Multiplying the Present Value by the Scale value yields a value indicated by the selected units.

Minimum Shows the minimum value for the setpoint.

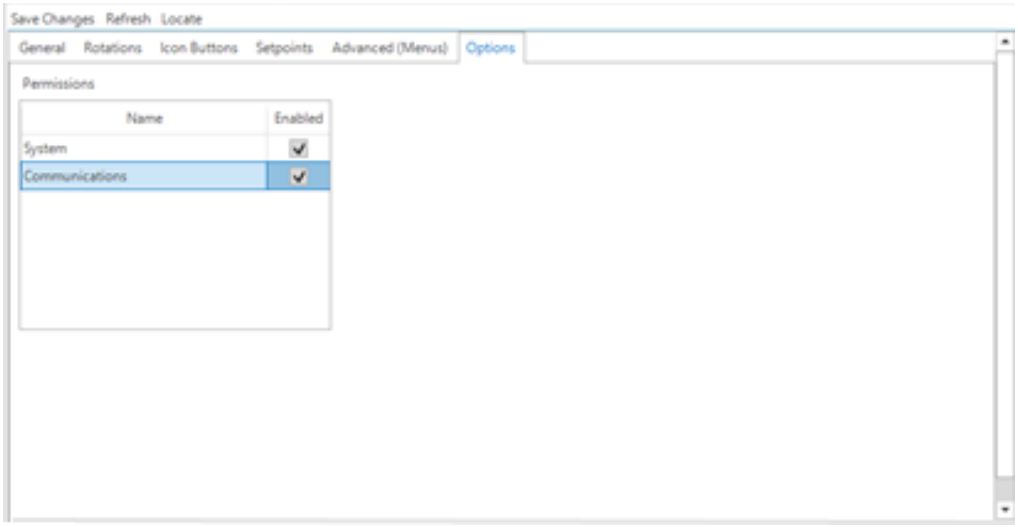
Maximum Shows the maximum value for the setpoint.

Options tab

The Options tab enables or disables access to the System and Communications menus in the Conquest FlexStat display. Disabling a menu prevents users from changing its properties from the FlexStat's touchscreen.

Note: If not enabled in the Permissions window, System or Communication options can be changed only in configuration software.

Illustration 37–7 Conquest FlexStat Options tab



Name Shows the System and Communication menu titles.

Enabled When selected, shows the System and Communication menus available in the Conquest FlexStat display.

Section 38: 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 is 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 562 and the Control Basic keyword *TBL* on page 329.

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 561.

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 Controls 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 38–1 Device table descriptions

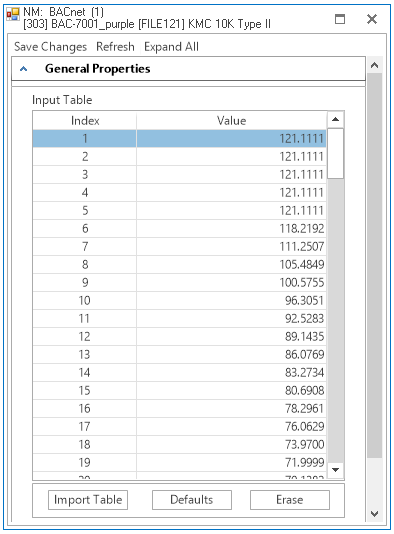
Device in table	Description
KMD 10K Type II	KMC Type II thermistors
KMD 10K Type III	KMC Type III thermistors
Airflow Sensor Table	BAC–7000 series VAV controllers

To open a BACnet Input table, see the topic *Configuring BACnet devices and objects* on page 410.

See the related topic *CB tables* on page 562 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 38–1 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 329 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.

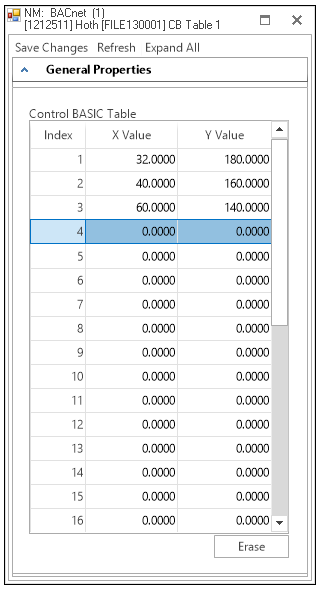
A table consists of 32 data pairs stored in two columns. Entries in the *X Value* column correspond to an input value, which is usually an analog input or value object. The *Y Value* entries are the result of the look-up function and are also stored in an analog object or variable.

- 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 410.

See also the keyword *TBL* on page 329 to use the table data in Control Basic.

Illustration 38–2 CB table



Section 39: Supported engineering units

KMC Connect supports the engineering units listed in this section.

Illustration 39–1 Analog units

%
%/sec
%Obscuration/ft
%Obscuration/m2
%RH
/hr
/min
/sec
amperes
amps/m
amps/m2
amps-sq-m
bars
BTU
BTU/hr
BTU/lb
BTU/lb-dry-air
candelas
candelas/m2
cm2
cm2
cm-mercury
cm-water
currency1
currency10
currency2
currency3
currency4
currency5
currency6
currency7
currency8
currency9
cycles/hr
cycles/min
Deg-Angular
DegC
DegC/hr
DegC/min
deg-days-C
deg-days-F
DegF
DegF/hr
DegF/min
DegK
DegK/hr
DegK/min
degrees-phase
delta-deg-F
delta-deg-K
farads
foot-canbdles
ft
ft/min
ft/sec
ft2
ft3
ft3/min
ft3/sec
grams/min
grams/sec
gr-water/kg-dry-air
hectoPa
henrys
hertz
horsepower
hours
Igal/min
Imperial-Gal
in
in2
in-mercury
inw
joules
joules/DegK
joules/kg-DegK
joules/kg-dry-air
joules/sec
KBTU
KBTU/hr
kg
kg/hr
kg/m3
kg/min
kg/sec
Khertz
KJ/kg
KJoules
Kjoules/DegK
Kjoules/kg-dry-air
KMH
KOhms
Kpa
KV
KVA
KVAR
KWH
KWH
KWH/ft2
KWH/m2
L/hr
L/min
L/sec
lbf/in2
lb-mass
lb-mass/hr
lb-mass/min
lb-mass/sec
liters
lixes
lumens
m/hr
m/min
m/sec
m2
m2/newton
m3
m3/hr
m3/min
m3/sec
mA
MBTU
meters
Meters/s/s
Mhertz
millibars
minutes
Mjoules
Mjoules/DegK
Mjoules/ft2
MJoules/kg-dry-air
Mjoules/m2
mm
mm/min
mm/sec
mm-mercury
MOhms
mOhms
months
MPH
msec
MV
MVA
MVAR
mVolts
MW
mwatts
MWH
newton
newton-m
newtons/m
newton-sec
no-units
ohm-meters
ohms
Pa
PF
ppb
ppm
psi/DegF
radians
radians/sec
RPM
sdays
sec/100
seconds
siemens
siemens/m
teslas
therms
ton-hours
tons
tons/hr
Tons-R
US-Gal
US-Gal/min
V/DegK
V/m
VA
VAR
Volts
watt-hours
watts
watts/ft2
watts/m/DegK
watts/m2
watts/m2/DegK
webers
weeks
years

Table 39–1 Binary unit pairs

Normal Inactive	Normal Active
OFF	ON
Stop	Start
Normal	Alarm
Closed	Open

Table 39–1 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 40: Reference to KMC Controls BACnet controllers

The information in the following table lists characteristics of the BACnet objects in KMC Controls BACnet controllers.

- For detailed specifications for each controller, see the installation and operation guide supplied with the controller.
- For information about connecting controllers to a building automation system, see the installation and operation guide supplied with the controller.
- For configuring the properties in an object, see the topic *Reference to BACnet objects* on page 435.

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 40–1 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

Table 40–1 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
Weekly schedule objects	8	8	8	8	8	8
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 40–2 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 40–3 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 Enrollment objects	10	

KMC Connect

Part VIII: KMDigital networks and points

Section 41: Configuring and connecting to KMDigital controllers and networks

The topics in this section cover connecting to a KMD network as well as configuring KMD points and a KMD network.

The following topics are basic procedures for working with KMDigital[1] networks, controllers and points in the Network Manager and configuration pages.

- *Before connecting to a KMDigital network* on page 573
- *Adding a KMDigital Tier 1 network* on page 574
- *Adding a KMDigital Tier 2 network* on page 579
- *Configuring KMD controllers and points* on page 581
- *Updating the Network Manager list* on page 583
- *Configuring a KMD service* on page 584
- *Configuring a KMD network* on page 586
- *Setting time for KMD networks* on page 587

These additional topics cover more advanced procedures for working with KMD controllers and points:

- *Reference to KMD points* on page 589
- *Configuring KMD controllers for alarms* on page 151

Before connecting to a KMDigital network

Before KMC Connect can configure KMD controllers, the KMD service must be running and the KMD network added to the KMD service. The service was automatically installed when KMC Connect was installed.

Before connecting to a KMD network, verify and have available the following items:

- The KMD service icon is present in the Network Manager list and that the icon is blue.
- Firewalls are open to network traffic used by the KMD ports listed in the topic *Firewalls and ports* on page 681.

---

[1]KMDigital is a proprietary line of digital building automation controllers manufactured by KMC Controls, Inc. 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 an EIA-485 network. KMDigital is often abbreviated as KMD.

Use one of the following procedures to add a KMDigital network to a KMC Connect site.

- *Adding a KMDigital Tier 1 network* on page 574
- *Adding a KMDigital Tier 2 network* on page 579

Related topics

- *Configuring KMD controllers and points* on page 581
- *Configuring a KMD service* on page 584
- *Configuring a KMD network* on page 586
- *Setting time for KMD networks* on page 587
- *Updating the Network Manager list* on page 583

Adding a KMDigital Tier 1 network

To connect KMC Connect to a KMD Tier 1 network, add a KMD network to the KMD service in the Network Manager list.

- For details on opening the Network Manager list, see *Using the Network Manager* on page 41.
- If a KMD service icon is not present and active in the Network Manager list, see *Adding a service* on page 44.
- Adding a Tier 1 network adds all KMD controllers in the KMD system to the Network Manager list.



Note: Before connecting a KMD controller to a network, use Hardware Configuration Manager to configure the controller with an address. Hardware Configuration Manager is available in the downloads area of the KMC Controls partners web site.

For a Tier 2 network, see the topic *Adding a KMDigital Tier 2 network* on page 579.

Adding a Tier 1 network connection

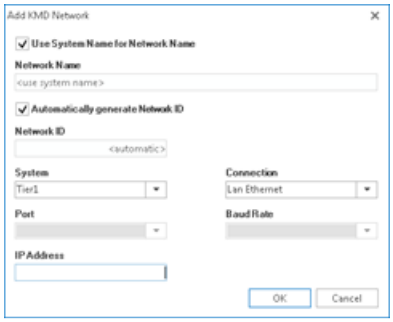
Use the following procedure to add a KMD network and Tier 1 controller to the KMD service. See also the procedure *Adding a Tier 1 serial connection* to connect to a Tier 1 controller with a serial connection.



Tip: You will need the network IP address of at least one of the Tier 1 KMD controllers in the KMDigital network for the following procedure.

To add a KMD Tier 1 network with a network connection, do the following:

1 In the Network Manager list, right-click the KMD Service icon and choose **ADD New KMD Network** from the shortcut menu. The **Add KMD Network** dialog opens.



2. To change the network name from the original KMDigital system name, do the following:
   a. Clear the **Use System Name for Network Name** check box.
   b. In **Network Name**, enter a new name for the network.
3. Under **System**, choose **Tier 1**.
4. Under **Connection**, choose **Lan Ethernet**.
5. Under **IP Address**, enter the address for one of the KMD Tier 1 controllers. Use one of the following methods:
   - If KMC Connect and the KMD controllers are on the same LAN, enter only the IP address of one of the Tier 1 controllers.
   - If a network address translation (NAT) router is connected between KMC Connectand the KMD network, add the characters `F:` or `R:` to the front of the public IP address. See the table *Network address translation prefixes* for a description of the prefixes.
   - If the IT department set up a URL name for one of the controllers, add the characters `F:` to the front of the URL. For example, enter `F:kmdnetwork.bigbuilding.com` for the IP address.
6. Click **OK** when finished.

The Network Manager list will add a network icon and then the list will fill with device icons. See the table *Network Manager KMD controller icons* on page 577 for a description of the KMD device icons.

Table 41–1 Network address translation prefixes

Prefix	Description
F	Full access—All Tier 1 controllers in the network status list are accessible. This is the typical method for connecting to a KMD network over the Internet when only one of the controllers has a public IP address.
R	Restricted access—Only the controller with the IP address in the translating router is accessible in the network status list.

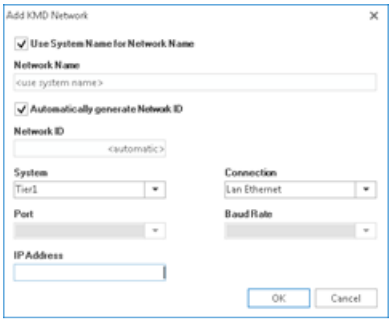
Adding a Tier 1 serial connection

See the illustration *KMD Tier 1 serial connection* on page 577 for connection details. Each type of controller requires a specific serial cable.

> **Tip:** Before you begin, verify the port number (COM1, COM2, etc.) of the serial port that the KMC Connect computer will use for the connection. Use the Windows Device Manager to find the port number.

To add a Tier 1 network with a direct serial connection, do the following:

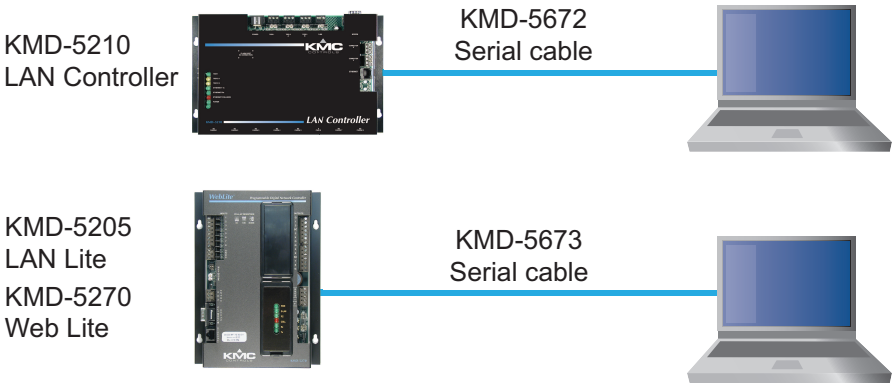
1. In the Network Manager list, right-click the KMD Service icon and choose **ADD New KMD Network** from the shortcut menu. The **Add KMD Network** dialog opens.



2. To change the network name from the original KMD system name, do the following:
   a. Clear the **Use System Name for Network Name** check box.
   b. In **Network Name**, enter a new name for the network.
3. Under **System**, choose **Tier 1**.
4. Under **Connection**, choose **Direct**.
5. Under **Port**, choose the port number for the serial connection (COM1, COM2, etc.).
6. Under **Baud Rate**, choose the same baud as the controller. Typically, this is 38,400 for Tier 1 controllers.
7. Click **OK** when finished.

KMC Connect will add a new network icon under the KMD Service icon and then the list will fill with device icons. For a description of the KMD controller icons, see *Icons for KMD controllers* on page 577.

Illustration 41–1 KMD Tier 1 serial connection



Icons for KMD controllers

Icons in the following table are added to a KMD network in the Network Manager list.

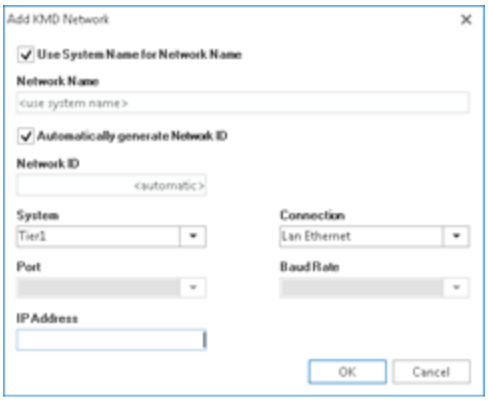
Table 41–2 Network Manager KMD controller icons

Tier 1	Tier 2	Status	Action
		All points in the controller are known to KMC Connect.	No action required. The controller is operating normally.
		KMC Connect has located the controller and is in the process of detecting points.	No action required. This icon will change to green once all points are detected.
		The controller has been detected but not all objects have been placed in the database.	Typically, no action is required. The icon will change to the icon with a gear.
		The controller 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 Refresh Device when the problem is corrected.
		Device is not available because of a system problem.	Verify that the KMD service is operating and licensed correctly.

Properties of the Add KMD Network dialog

Properties in the Add KMD Network dialog set the connection method between a KMDigital network and the KMD service.

Illustration 41–2 Add KMD Network dialog



Use System Name for Network Name When selected, KMC Connect will automatically assign the original KMDigital system name as the KMC Connect network name. To use a different name, clear the **Use System Name for Network** check box.

Network Name A descriptive label of the network.

System Selects the type of KMD network to which KMC Connect will connect. The choices are Tier 1 or Tier 2.

Connection Depending on the type of network, the Connection list will have the following choices:

Tier 1 Lan Ethernet, Direct, or Modem.

Tier 2 has only Direct as a choice.

Port Selects the serial port (Com 1, Com 2, etc.) on the computer running KMC Connect.

Baud Rate Sets the speed of the serial connection. Set Baud Rate to the same speed as the controllers on the network.

IP Address The IP address of a KMD Tier 1 controller on the network. The address may take one of the following forms.

- If KMC Connect and the KMD controllers are on the same LAN, enter the IP address for one of the Tier 1 controllers.
- If a network address translation (NAT) router is connected between KMC Connectand the KMD network, add the characters `F:` or `R:` to the front of the public IP address. See the table *Network address translation prefixes* for a description of the prefixes.
- The IT department may set up a URL name for one of the controllers. To use a URL, add the characters `F:` to the front of the URL. For example, enter `F:kmdnetwork.bigbuilding.com` for the IP address.

Automatic Network ID Select the **Automatic Network ID** check box when adding a new KMD network. Clear the check box to change the network number that was automatically assigned by KMC Connect.

Network ID Use only to change the Network ID number that was automatically assigned by KMC Connect.

Related topics

- *Before connecting to a KMDigital network* on page 573
- *Adding a KMDigital Tier 2 network* on page 579
- *Configuring KMD controllers and points* on page 581
- *Updating the Network Manager list* on page 583
- *Configuring a KMD service* on page 584
- *Configuring a KMD network* on page 586
- *Setting time for KMD networks* on page 587

Adding a KMDigital Tier 2 network

To connect a KMC Connect site to KMD Tier 2 controllers, add a new KMD network to the KMD service in the Network Manager list.

Note: Before connecting a KMD controller to a network, use Hardware Configuration Manager to configure the controller with an address. Hardware Configuration Manager is available in the downloads area of the KMC Controls partners web site.

Connect to the computer running KMC Connect with one of the methods shown in the diagrams *Connecting with a KMD-5559* on page 580 and *Connecting with a KMD-5576* on page 581. Either method requires specific cables for the connection. Additional cable details are available in the instructions supplied with the KMD-5559 and KMD-5576. When connecting with a KMD-5559, use a third-party USB-to-serial adapter if the computer running KMC Connect does not have a 9-pin serial connector.

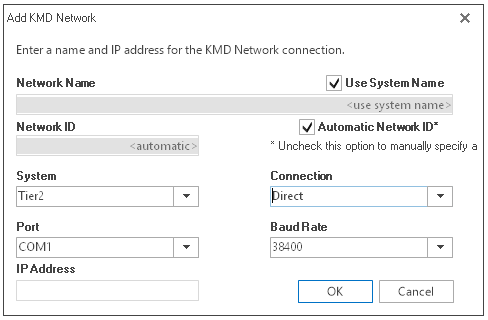
Note: KMC Connect does not process KMDigital alarms stored in a KMD-5559.

When KMC Connect is connected directly to a Tier 2 network, only the controllers on that network are added to the Network Manager list.

Tip: Before you begin, verify the port number (COM1, COM2, etc.) of the serial port that the KMC Connect computer will use for the connection. Use the Windows Device Manager to find the port number.

To add a Tier 2 network with a serial port connection, do the following:

1 In the Network Manager list, right-click the KMD Service icon and choose **ADD New KMD Network** from the shortcut menu. The **Add KMD Network** dialog opens.



2 To change the network name from the KMD system name, do the following:

   a. Clear the **Use System Name** check box.

   b. In **Network Name**, enter a new name for the network.

3 Under **System**, choose **Tier 2**.

4 Under **Port**, choose the port for the serial connection (COM1, COM2, etc.).

5 Under **Baud Rate**, choose the baud of the network.

   - For a KMD-5559 connection use 19,200.
   - For a KMD-5559-2 use 9,600.
   - For other connections, enter the same baud as the controllers on the network.

6 Click **OK** when finished.

KMC Connect will add a new network icon and device icons to the Network Manager list. This process may take several minutes. Only the controllers connected to the Tier 2 network are added to the Network manager list. For a description of the KMD device icons, see *Network Manager KMD controller icons* on page 577.

Illustration 41–3 Connecting with a KMD-5559

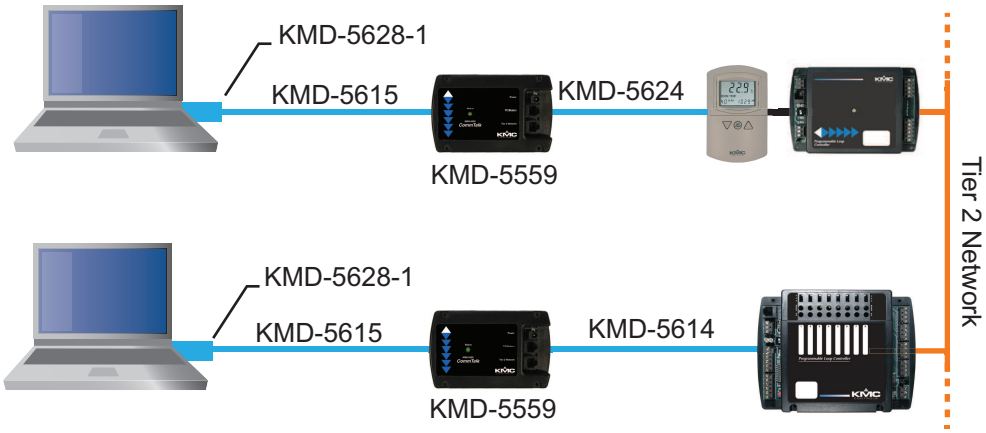
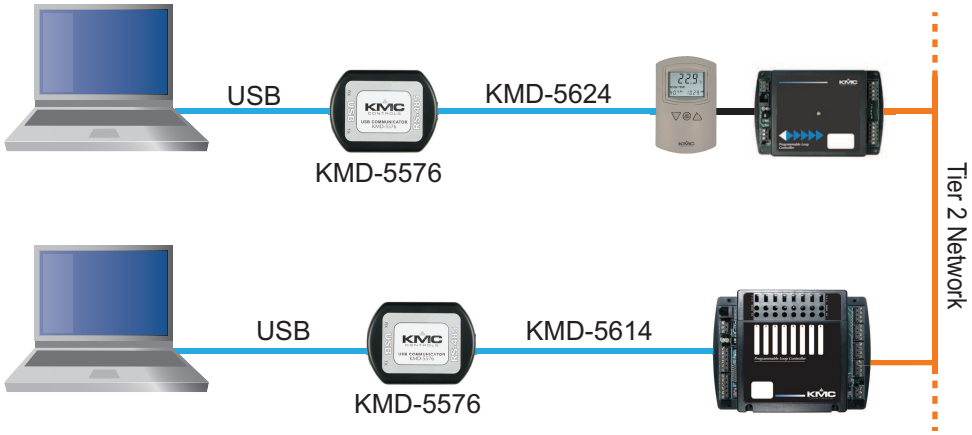


Illustration 41–4 Connecting with a KMD-5576



Configuring KMD controllers and points

To configure a KMD controller or point, the controller must have a device icon in the Network Manager list. If a device icon is not in the Network manager list, see *Configuring a KMD network* on page 586. For controllers in the Network Manager list you may use any 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 from the Resource Manager.
- By copying the configuration from a similar controller or point that is in the Network Manager list.

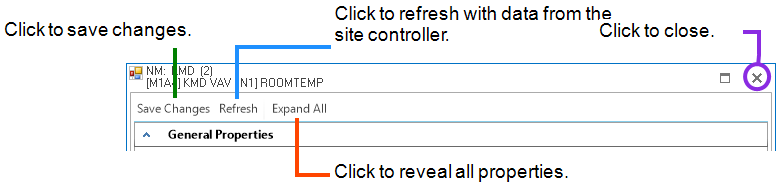
To open a configuration tab for a KMD controller or object, do the following:

1. In the Network Manager list, click ⊞ or ⊟ to expand or collapse the list to locate a specific KMD controller or point.
   - The controller device icon will always be in the list under a KMD service icon and network icon .
   - A KMD point is in a folder under a device icon. Each type of point has a unique icon.
2. Once the device or point is located, double-click the object or point 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 41–5 Tab in KMD configuration page



Copying configurations

The configuration of a device or object may be copied from either of two sources:

- Directly from another connected controller in the Network Manager.
- From a backup file in the Resource Manager.

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 controller or point.

2 Hold ALT and drag the configuration to the object, folder of points or device.

When dragging a configuration file from the Network Manager list, the following actions are supported:

- Dragging a point configuration file onto a single point of similar type. For example, drag only input point configurations onto other input points.
- Dragging a controller configuration file onto another controller that contains points of similar type. A dialog opens with which you can select the specific points to restore.
- Dragging a point onto a network that includes devices with similar objects.
- Dragging a folder of points onto a similar folder. The points are matched by their point instance number to the points in the Network Manager list.

Related topics

- *Before connecting to a KMDigital network* on page 573
- *Adding a KMDigital Tier 1 network* on page 574
- *Adding a KMDigital Tier 2 network* on page 579
- *Updating the Network Manager list* on page 583

- *Configuring a KMD service* on page 584
- *Configuring a KMD network* on page 586
- *Setting time for KMD networks* on page 587

Updating the Network Manager list

To update the KMD controller and network information in the Network Manager list, use Regenerate KMD Device, Regenerate Device List or Refresh Device List.

- Refreshing the device list updates the database with information from existing controllers and adds any new controllers that have been connected to the network.
- Regenerating either a device or the entire Network Manager list removes devices from both the database and the Network Manager list. KMC Connect then restores only the devices connected to the network.

To refresh the Network Manager list:

1. In the Network Manager list, click ⊞ or ⊟ to expand or collapse the list to locate the network icon for a specific KMD network.
2. Right-click the icon and then choose **Refresh Device list**. As new controllers are discovered, the are added to the list.

To regenerate a single device:

1. In the Network Manager list, click ⊞ or ⊟ to expand or collapse the list to locate the device icon for a specific KMD controller.
2. Right-click the icon and then choose **Regenerate KMD Device**.
3. The icon will be removed from the list and then return as a yellow icon. Regeneration is complete when the icons change to green.

To regenerate the network:

1. In the Network Manager list, click ⊞ or ⊟ to expand or collapse the list to locate the network icon for a specific KMD network.
2. Right-click the icon and then choose **Regenerate Device List**.
3. All network icons are removed and then return as yellow icons. Regeneration is complete when the icons change to green.

Related topics

- *Before connecting to a KMDigital network* on page 573
- *Adding a KMDigital Tier 1 network* on page 574
- *Adding a KMDigital Tier 2 network* on page 579
- *Configuring KMD controllers and points* on page 581
- *Configuring a KMD service* on page 584

- *Configuring a KMD network* on page 586
- *Setting time for KMD networks* on page 587

Configuring a KMD service

Use the KMD Service configuration tab to view or change settings for the KMD service.

1. Locate the KMD Service icon in the Network Manager list.
2. Right-click the KMD 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** in the tab of the configuration page.

Set KMD Service as Default

The default service in TotalControl can be set to either BACnet, KMD, or OPC. TotalControl graphic elements with a service ID of 0 will use this service.

To set the KMD Service as the default:

1. Locate the KMD Service icon in the Network Manager list.
2. Right-click the KMD Service icon and choose **Set as Default** from the shortcut menu.



Note: "Set as Default" is unavailable if the service is already the default.

3. When changes are complete, click **Save Changes to Service** in the tab of the configuration page.

General Properties

Items under General Properties are usually set up when the KMD service was added to the Network Manager list.

Service Name A descriptive name of the service to display in the Network Manager list.

Service ID The Service ID number that was automatically assigned by KMC Connect.

Service Host Address Not applicable to KMC Connect.

Service Host TCP Port Not applicable to KMC Connect.

Update System Group File The KMD service maintains a list of all the System Groups in each controller in the system. Selecting the Update System Group File check box enables KMC Connect to update the list as changes are made. Clearing the check box prevents changes to the list.

See also the related topic *System Groups* on page 613.

Settings and Statistics

The entries in the Settings and Statistics area either set or display operational information about the KMD service.

Maximum Response Staleness Sets the age of the data. If—at the time data is requested from a point—the data in the database is older than the value of *Maximum Response Staleness*, KMC Connect will retrieve new data from the controller. If the data is newer than the value in *Maximum Response Staleness*, KMC Connect displays the value stored in the database.

Time Synchronization Interval Sets the interval at which KMC Connect will updated the time on the connected network. The time sent to the network is the time maintained in the computer on which the KMD service is running.

Monitor Statistics Select to automatically refresh data in the Incoming Requests, Pending Requests and Completed Requests fields.

Incoming Requests The number of requests the service has received from KMC Connect. During device discovery, activity will peak and then subside as discovery is completed.

Pending Requests The number of unanswered requests that the service has sent to the networks. During discovery the rise and fall of this value will somewhat lag the incoming requests rise and fall.

Completed Requests The number of requests sent to networks for which the service received a response. This is a cumulative value since the service was last started. Selecting the *Monitor Statistics* check box will periodically initiate requests of the service, which increases the *Completed Requests* by one(1) each time a request is made.

Licensing

This area contains settings and information about licensing for the service.

License Status This is a list of the features for the service that are authorized by the license key. This includes also the number of devices or controllers that are authorized by the license.

License Address This is the location of the license key. If the key is plugged into the computer on which the service is running, the entry is `(local)`. To use a license key on another computer on the network, enter either the computer name or IP address of the computer.

License Port This is the UDP network port for communications between the service and the KMC License Service. Do not change this unless instructed by KMC Controls Technical Support. The default entry is 27605.

Related topics

- *Before connecting to a KMDigital network* on page 573
- *Adding a KMDigital Tier 1 network* on page 574
- *Adding a KMDigital Tier 2 network* on page 579
- *Configuring KMD controllers and points* on page 581

- *Updating the Network Manager list* on page 583
- *Configuring a KMD network* on page 586
- *Setting time for KMD networks* on page 587

Configuring a KMD network

A KMD network is configured at the time it is added to the Network Manager list. Once added to the list only the network name can be changed. To change other properties of the network, see *Adding a KMDigital Tier 1 network* on page 574 or *Adding a KMDigital Tier 2 network* on page 579.

- If a KMD network is not present, see *Adding a KMDigital Tier 1 network* on page 574 or *Adding a KMDigital Tier 2 network* on page 579.
- To change the properties of a KMD service, see *Configuring a KMD service* on page 584
- If a KMD service is not in the Network Manager list, see *Adding a service* on page 44.

To change the name of an existing KMD network, do the following:

1. Expand or collapse the Network Manager list to locate the network name.
2. Right-click the network icon and choose **Configure KMD Network**.
3. In the configuration page, make the changes to the network.
4. When changes are complete, click **Save** in the tab at the top of the configuration page.

General Properties

Network Name A descriptive label of the network.

Network ID The Network ID is assigned by KMC Connect and cannot be changed.

IP Address The IP address of a KMD Tier 1 controller on the network. This can only be changed with Hardware Configuration Manager. It cannot be changed in KMC Connect.

Tier 1 Network Status

The status properties apply to the Tier 1 network to which the controller is assigned.

To see the Tier 2 Network Status, see the topic *Device points* on page 592.

Online Indicates *TRUE* if the controller is currently on-line.

Running Program Indicates *TRUE* if one or more Control Basic programs are running in the controller. See also the topic *About Control Basic scans* on page 218.

Version The revision level of the firmware in the controller.

Scan Rate This is the rate that all Control Basic programs in the controller are running. Typical scan rates range from 5 to 50 scans per second. See also the topic *About Control Basic scans* on page 218.

Network Input Point The total number of want-points a controller is requesting from other controllers. See also the topic *Transferring values with KMD want-points* on page 240.

Network Output Point The total number of want-points a controller is making available to other controllers.

Remaining memory The amount of free memory, in bytes, available for programs, trend logs, runtime logs, etc.

Advanced right-click commands

The following commands are available only by right-clicking the network icon in the Network Manager list.

Read System Groups Refreshes the points in all of the System Groups in all controllers on the network. See also the procedure *Reading the System Group list* on page 614.

Clear Want Points Clears all want-points from the controllers on the network. Clear the want-points when a controller has exceeded its maximum number of points. See *Transferring values with KMD want-points* on page 240 for detail.

Related topics

- *Before connecting to a KMDigital network* on page 573
- *Adding a KMDigital Tier 1 network* on page 574
- *Adding a KMDigital Tier 2 network* on page 579
- *Configuring KMD controllers and points* on page 581
- *Updating the Network Manager list* on page 583
- *Configuring a KMD service* on page 584
- *Setting time for KMD networks* on page 587

Setting time for KMD networks

Use the KMD Service configuration tab to set the interval at which KMC Connect will update the time in the controllers connected to the KMD networks. The time sent to the networks is the time of the computer on which the KMC Connect is running.

1. Locate the KMD Service icon in the Network Manager list.
2. Right-click the KMD Service icon and choose **Configure Service** from the shortcut menu.
3. Under **Time Synchronization Interval**, choose an update interval:
   - Never
   - Hourly
   - Daily
   - Weekly
   - Monthly

4 Click **Save Changes to Service** at the top of the page.



Note: The actual KMDigital network time is maintained in the controller with the lowest Tier 1 panel number connected to the network. See the topic *Device points* on page 592 to set the time in a KMDigital device.

Related topics

- *Before connecting to a KMDigital network* on page 573
- *Adding a KMDigital Tier 1 network* on page 574
- *Adding a KMDigital Tier 2 network* on page 579
- *Configuring KMD controllers and points* on page 581
- *Updating the Network Manager list* on page 583
- *Configuring a KMD service* on page 584
- *Configuring a KMD network* on page 586

Section 42: Reference to KMD points

The following topics are a reference to the features and characteristics of points in KMD series controllers.

KMC Connect supports standard points in KMD series controllers.

- To locate a KMD controller or point on a KMD network, see *Using the Network Manager* on page 41.
- To configure the points in a KMD controller, see *Configuring and connecting to KMDigital controllers and networks* on page 573.

This section describes the following objects:

- *Annual Schedule points* on page 589
- *Array points* on page 591
- *Device points* on page 592
- *Input points-analog* on page 594
- *Input points-binary* on page 595
- *InterProtocol points* on page 596
- *Configuring a NetSensor for KMDigital controllers* on page 599
- *Output points-analog* on page 605
- *Output points-binary* on page 607
- *PID loop controllers* on page 608
- *Control Basic programs* on page 611
- *Runtime Log points* on page 611
- *System Groups* on page 613
- *Tables* on page 614
- *Trend Log points* on page 615
- *Variable points-analog* on page 616
- *Variable points-binary* on page 617
- *Weekly Schedule points* on page 618

Annual Schedule points

Use Annual Schedule points to override the normal schedule in a Weekly Schedule point on special days during a year. When the calendar maintained in the controller reaches one of the days designate by the Annual Schedule point, the point changes to *Active* (*On, 1,* etc.).

An Annual Schedule point can be used to override a normal daily schedule in a Weekly Schedule point. The state of a schedule can also be tested with Control Basic as shown in the following example.

```
A = AS1
Override = AS2
```

General Properties

Name The 8-character description of the point. **Name** must be unique within the KMD controller that maintains it. The set of characters entered for **Name** must be printable characters.

Description A 20-character description of the annual schedule. **Description** must be unique within the KMD controller that maintains it. The set of characters entered for **Description** must be printable characters.

Present Value Indicates whether the schedule is *Active (On)* or *Inactive (Off)*.

Manual When the Manual check box is selected, the present value will remain as displayed until changed by a user. When the check box is clear, the value of the schedule will change based on the days selected.

Annual Schedule Dates

To add to or remove days from an Annual Schedule point, click the day on the calendar. Enabled dates appear highlighted. When the highlighted day begins, the present state of the Annual Schedule changes to *Active (On).*

Alarm Routing Properties

Alarm classes designate a group of operators that receive notifications of alarms and events originating from this point. Select the KMC Connect alarm class routing from the list. See the topics *An overview of notifications, alarms, and events* on page 133, *Emailing operator alarms* on page 171, and *Alarm classes for KMD controllers* on page 153.

Related topics

- *Configuring KMD Weekly Schedule points* on page 190
- *Configuring KMD Annual Schedule points* on page 191
- *Weekly Schedule points* on page 618
- *About KMD Schedules* on page 189

Array points

An array is a special kind of table. It contains only one column and up to 128 rows which are also referred to as elements. The elements of the array can take on any numerical value and each element value can be recalled or updated at any time using Control Basic. An example of an array would be the storing of the highest temperature every day of the month.



Note: Available only in KMD Tier 1 controllers.

See the related topic *Configuring KMD controllers and points* on page 581.

Array Name The 8-character description of the point. the Array Name must be unique within the KMD controller that maintains it. The set of entered characters must be printable characters.

Array Length The total number of elements the array can contain. The array can be set to hold up to 128 elements. The elements of the array can be assigned either manually or from a Control Basic program.

Index The first column displays the element number of the array. This number is used to reference data in the value column. The number of elements in an array is set in Array Length.

Value The actual data can be manually entered in the *Value* column or a Control Basic program can read or write the data. When specifying an element within an array, use the form *AYN [x]*.

- *N* is the array number
- *x* is the element number within the array

Both *N* and *x* must be whole numbers.

The following example shows how to set the value of a variable from an array element:

```
REM Set Variable 6 to the value of Array 2, element 11
VAR6=AY2[11]
```

This example shows how to set the values of array AY1:

```
IF+ 1.VAR1 < 55 THEN GOSUB 30
END
IF X > 127 THEN X = 0 : REM Line 30
X = X + 1
1.AY1[ X ] = 1.VAR2
X = X + 1
1.AY1[ X ] = 1.VAR1
RETURN
```

This program records the position value of the mixed air damper (variable VAR2) and the time of day when the mixed air temperature (variable VAR1) falls below 55 degrees. Local variable *X* selects the element where the data is to be stored. Time values are recorded in hours (12:30 PM =1250). The array should be configured for 128 elements. Once the array is full, new data will be written over the oldest data.

Device points

To configure a KMD device, do the following:

1. In the Network Manager list, locate a Tier 1 device or Tier 2 device icon.
2. Do one of the following:
   - Double-click the icon.
   - Right-click the device icon and choose **Configure KMD Device** from the shortcut menu. The device configuration tab opens.
3. Make changes and then click **Save Changes** at the top of the tab.

General Properties

Panel Name Enter an 8-character description of the device.

Plus Mode Select the **Plus Mode** check box to set the PID control loop calculation frequency. This is not available on all models.

DayLight Saving Time Use to set start and end dates to observe daylight savings time. Select **Disable** in locations where daylight saving time is not observed.

Device Status A view only property.

Firmware Version The firmware series.



Note: Firmware Version is not the same as the firmware build number.

Date and Time (Tier 1 controllers only) Use to directly enter the time and date in the controller.

The Tier 1 controller with the lowest controller number maintains the time for the entire KMD network.

Build Number A view only property. The build number is the released version number of the firmware.

KMD Model Number A view only property.

Build Time and Date View only properties. Does not apply to all models.

MAC Address The MAC address uniquely identifies the controller on the network. It is assigned by KMC Controls and cannot be changed.

Tier 2 Network Status

Properties for Tier 2 Network Status apply only to the Tier 2 networks directly connect to the Tier 1 controller

To see the Tier 1 Network Status, see the topic *Configuring a KMD network* on page 586.

Online Indicates *TRUE* if the controller is currently on-line.

Running Program Indicates *TRUE* if one or more Control Basic programs are running in the controller.

Version The revision level of the firmware in the controller.

Scan Rate This is the rate that all Control Basic programs in the controller are running. Typical scan rates range from 5 to 50 scans per second. See also the topic *About Control Basic scans* on page 218.

Network Input Point The total number of want-points a controller is requesting from other controllers. See also the topic *Transferring values with KMD want-points* on page 240.

Network Output Point The total number of want-points a controller is making available to other controllers.

Custom Units

Custom Units are user defined units of measure for use with inputs, outputs, variables, arrays and tables. A Custom Unit defined in one controller is automatically distributed to, and stored in all controllers on the same network.

Analog Use Analog Custom Units with analog output and variable points. Analog Custom Units are limited to five characters.

Digital On Use the Binary On Custom Unit with binary points that are in the Active state. Digital On Custom Units are limited to eleven characters.

Digital Off Use the Binary Off Custom Unit with binary points that are in the Inactive state. Digital Off Custom Units are limited to eleven characters.

Alarm Routing Properties

Alarm classes designate a group of operators that receive notifications of alarms and events originating from this point. Select the KMC Connect alarm class routing from the list. See the topics *An overview of notifications, alarms, and events* on page 133, *Emailing operator alarms* on page 171, and *Alarm classes for KMD controllers* on page 153.

Additional right-click commands

The following commands are available only by right-clicking a Tier 1 device  or Tier 2 device  icon in the Network Manager list.

Erase Master Password Removes the master KMDigital password from all controllers on the same KMDigital network.



Caution: The master password cannot be restored unless all programming is erased from the controller with the HCM configuration program. Before disabling the master password, verify that all user defined passwords are known and recorded and that at least one user defined password has level 6 access.

Initialize Print Buffer Deletes all print jobs in the KMD–5210 print buffer.



Note: For KMD–5110 MultiNet controllers only.

Read System Groups Refreshes the points in the System Groups. See also the procedure *Reading the System Group list* on page 614.

Clear Panel Erases all programming, point configuration, labels, and descriptions from the controller. Using Clear Panel does not change settings made with HCM. The configuration for panel number, baud, and last panel remain unchanged.

Clear Want Points Clears all want-points from the controllers on the network. Clear the want-points when a controller has exceeded its maximum number of points. See *Transferring values with KMD want-points* on page 240 for detail.

Input points-analog

An analog input point represents the characteristics of an analog signal at the input of a controller.

General Properties

Name The 8-character description of the point. **Name** must be unique within the KMD controller that maintains it. The set of characters entered for **Name** must be printable characters.

Description A 20-character description of the device connected to the point. **Description** must be unique within the KMD controller that maintains it. The set of characters entered for **Description** must be printable characters.

Binary State When the Binary State check box is clear, the input is set as an analog input. When the check box is selected, the input is a binary input.

Value The current level, quantity, or state of the point.

Units Choose the unit of measure and scale factor for the input signal. For a list of units, see *Supported KMD controller engineering units* on page 621.

Manual Override Indicates the input is either in automatic or manual mode. When **Manual Override** is selected, the input value will remain as displayed until changed by an operator. When the check box is clear, the point will display the value at the input of the controller.

Decommissioned Indicates the input value has been out of range for more than two minutes. If the point is functioning normally, this check box is clear. **Decommissioned** is valid only for inputs configured for a KMC thermistor or table.

See the keyword *DECOM* on page 270 to detect the state of **Decommissioned** in a Control Basic program.

Average Sets the number of samples that are averaged together to calculate the displayed value. A sample is taken on each scan.

Calibration (Analog Inputs points only) 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).

Alarm Routing Properties

Alarm classes designate a group of operators that receive notifications of alarms and events originating from this point. Select the KMC Connect alarm class routing from the list. See the topics *An overview of notifications, alarms, and events* on page 133, *Emailing operator alarms* on page 171, and *Alarm classes for KMD controllers* on page 153.

Related topics

- *Configuring KMD controllers and points* on page 581
- *Input points-binary* on page 595 to change an analog input to a binary input

Input points-binary

A binary input point represents the characteristics of a binary signal at the input of a controller.

General Properties

Name The 8-character description of the point. **Name** must be unique within the KMD controller that maintains it. The set of characters entered for **Name** must be printable characters.

Description A 20-character description of the device connected to the point. **Description** must be unique within the KMD controller that maintains it. The set of characters entered for **Description** must be printable characters.

Binary State When selected, the input is a binary input. When the Binary State check box is clear, the input is set as an analog input.

Value The current state of the point.

Units Select the unit of measure to match the input device. In the scroll box, the digital units of measure are shown as a pairs. The first unit is the normal state. When referencing digital inputs in Control Basic, the left hand state is *False* (0) and the right hand state is *True* (1). To reverse the action, select the **Invert Polarity** check box.

For a list of available units, see *Reference to KMD points* on page 589.

Invert Polarity Sets the relationship between the physical state of the input and the logical state represented by **Value**.

Table 42–1 Input object polarity relationships

Invert Polarity	Voltage at input	Dry contact with pull-up	Example Unit	Text displayed
Clear	0	Closed	Off On	Off
Clear	5	Open	Off On	On
Selected	0	Closed	Off On	On
Selected	5	Open	Off On	Off

Manual Override Indicates the input is either in automatic or manual mode. When **Manual Override** is selected, the input value will remain as displayed until changed by an operator. When the check box is clear, the point will display the value at the input of the controller.

Decommissioned Indicates the input value has been out of range for more than two minutes. If the point is functioning normally, this check box is clear. **Decommissioned** is valid only for inputs configured for a KMC thermistor or table.

To detect the state of **Decommissioned** in a Control Basic program, see the keyword *DECOM* on page 270.

Alarm Routing Properties

Alarm classes designate a group of operators that receive notifications of alarms and events originating from this point. Select the KMC Connect alarm class routing from the list. See the topics *An overview of notifications, alarms, and events* on page 133, *Emailing operator alarms* on page 171, and *Alarm classes for KMD controllers* on page 153.

Related topics

- *Configuring KMD controllers and points* on page 581
- *Input points-analog* on page 594 to change a binary input to an analog input

InterProtocol points

InterProtocol points are applicable only to licensed controllers only. Use InterProtocol Points to associate a register in a Modbus[1] device with a variable in the KMD controller. Each InterProtocol Point transfers one value between the Read Point and the Write Point in the Read/Write Points area.

See the related Control Basic keyword *MODBUSTRANSFER* on page 295.

[1]A protocol developed to establish master-slave/client-server communication between industrial devices. 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.

General Properties

Description Enter a description of the data passed between a KMD controller and the Modbus controller. Description is for notation only and is not required for operation.

Mode Sets the update mode for each read point and write point pair.

- **Disabled**–The pair will not be processed.
- **One Time**–The controller will update the pair only when Save Changes is clicked. The value for the pair will not be processed after that.
- **Continuous**–The pair is processed approximately once every 10 seconds.

Value The current level, quantity, or state of the point.

Read/Write Points

Read Point Read Point sets the KMD point or Modbus register whose value will be transferred to Write Point. Read Point can be also set to a fixed value. The timing of the transfer is set by Mode.

Set Read Point to either KMD, Modbus or Fixed Point.

- If Read Point is set to KMD, enter the mnemonic for a KMD point. Use standard Control Basic mnemonics as described in the topic *Mnemonics for KMD controllers* on page 239.
- If Read Point is set to Modbus, see Modbus Point for details on specifying a Modbus register.
- If Read Point is set to Fixed Value, enter any valid number.

Read Status Reports the success or failure of reading the value in Read Point.

Write Point Write Point is the KMD Point or Modbus register that will receive the transfer from Read Point.

- If Write Point is set to KMD, enter the mnemonic for a KMD point. Use standard Control Basic mnemonics as described in the topic *Mnemonics for KMD controllers* on page 239.
- If Write Point is set to Modbus, see Modbus Point for details on specifying a Modbus register.

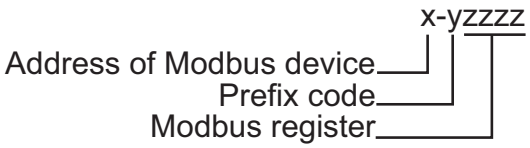
Writes Status Reports the success or failure of updating the KMD point or Modbus register in Write Point.

Modbus Point Modbus Point is active only when Read Point or Write Point is set to Modbus.

Point String Enter Modbus registers as shown in the illustration *Modbus register format* on page 598. Function codes are listed in the table *Supported Modbus function numbers and prefix codes* on page 598.

Note: Modbus manufacturers typically list the prefix code with the register number in equipment guides. Verify in the equipment guide exactly how the register number is formatted.

Illustration 42–1 Modbus register format



Supported Modbus function numbers and prefix codes

Function		Prefix code	Action	Description
0	00 hex	0	Read	Discrete output coil
1	01 hex	1	Read	Discrete output coils
3	03 hex	4	Read	Analog output holding registers
4	04 hex	3	Read	Analog input registers
6	06 hex	4	Write Single	Analog output holding register
15	0F hex	0	Write multiple	Discrete output coils

Data Type Data Type sets the type of data stored in the register.

- **Unsigned Integer**–For whole numbers between 0 and 65535.
- **Signed Integer**–For whole numbers between -32768 and 32767
- **Bit String**–Use for digital points and status registers.
- **Float**–Must use for 32-bit registers.

Data Format Data Format sets the order in which the parts of the data are sent first. This is also referred to as *byte ordering* and *word ordering*.

Integer data types have only byte ordering (LSB and MSB) to select. To set up Data Format for integers choose one of the following:

- Choose **MSB,LSB** to send the most significant or high byte first.
- Choose **LSB,MSB** to send and store the least significant or low byte first.

Float data types have both word ordering (MSW and LSW) and byte ordering (MSB and LSB) to set up. To set up Data Format for floating point values, choose one of the following:

- Choose **MSW, LSB** or **MSW, MSB** to send the most significant data word first followed by either the least significant (LSB) or most significant byte (MSB).
- Choose **LSW, LSB** or **LSW, MSB** to send the least significant data word first followed by either the least significant (LSB) or most significant byte (MSB).

Multiplier and Offset Values for Multiplier and Offset are used for scaling. Both are provided by the manufacturer of the Modbus equipment.

Number of Registers Choose 16 bit or 32 bit. If Float is selected in Data Type, set Number of Registers to 32 bits.

Configuring a NetSensor for KMDigital controllers

The KMD-1160, KMD-1180, KMD-1260, and KMD-1280 series NetSensors connect directly to any of the following KMDigital controllers:

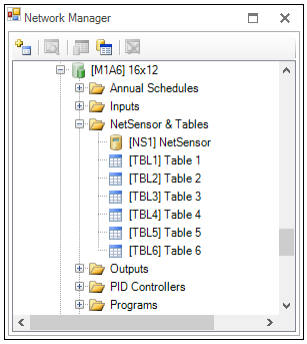
- KMD-5800 series
- KMD-7000 series
- KMD-7400 series

The NetSensors are wall mounted display and sensor units that consist of an LCD display, a thermistor, up to seven buttons, and optional humidity and motion sensors. Additional information for the NetSensor can be found in the following documents.

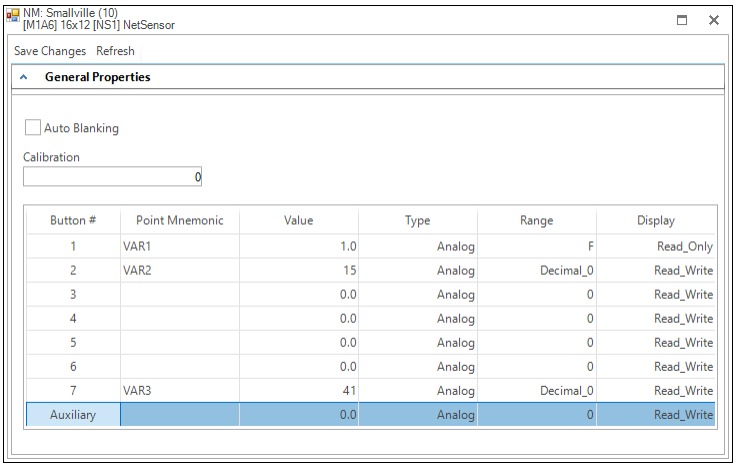
- In the installation guide for the NetSensor.
- In the Control Basic topic *NETSENSOR-STATUS* on page 300.

To configure a KMD series NetSensor object, do the following:

1. In the Network Manager list, click ⊞ or ⊟ to expand or collapse the list to open the NetSensor & Table folder.



2 Double-click the NetSensor icon to open the NetSensor object. The NetSensor tab opens.



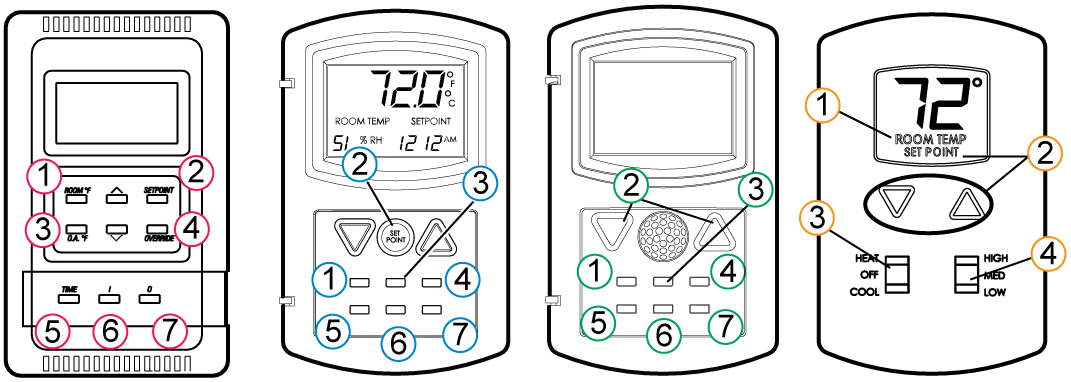
3 As required, configure each of the buttons. See *Button configuration examples* on page 600 for suggestions on configuration.

4 When finished, click **Save Changes**.

Button configuration examples

Programming a KMD series NetSensor consists of assigning an object—typically a KMD variable point—to one of the buttons on the NetSensor. The value of the point is then displayed by the NetSensor and controlled with the assigned button. See NetSensor buttons for a description of the available button functions.

Illustration 42–2 NetSensor models and button layouts



Use the following examples to set up and program a NetSensor to display room temperature, humidity, a setpoint and time.

Caution: When displaying the value from the internal temperature sensor, always associate Button 1 only with a variable. Associating Button 1 with an input or output will result in improper operation.

Button 1—internal temperature sensor Button 1 and the space temperature sensor inside of the NetSensor is usually associated with variable VAR1.

1. Under **Point Mnemonic**, choose the mnemonic of the variable that will store temperature data.
2. Set **Type** to **Analog**.
3. Set **Range** to **F** or **C**.
4. Set **Display** to **Read Only**. Operators can then view the room temperature by pressing Button 1 but cannot change the value.

Button 1—controller points To manage or display a point value from Button 1, configure as follows:

1. Under **Point Mnemonic**, choose the mnemonic of the point to associate with Button 1.
2. Set **Type** to **Analog** or **Decimal**.
3. Set **Range** to **0, 1** or **2 Decimal**. Do not set range to *F* or *C* unless controlling the internal temperature sensor.
4. Set **Display** to **Read** or **Read/Write**.
   - If set to **Read**, operators can view the value associated with the Button but cannot change it.
   - If set to **Read/Write**, operators can change the value of the point by first pressing Button 1 and then pressing the up or down arrow buttons.

Setpoint Button 2 is usually associated with Variable VAR2.

1. Under **Point Mnemonic**, choose the mnemonic of the variable that will store the setpoint value.
2. Set **Type** to **Analog**.
3. Set **Range** to **0, 1** or **2 Decimal**.
4. Set **Display** to **Read/Write**.

   Operators can view and change the setpoint by first pressing Button 2 and then pressing an up or down arrow button.
5. Write a Control Basic to control equipment based on the conditions of the setpoint.

   ```
   IF VAR2 < VAR1 THEN START OUT6
   ```

Humidity (Humidity equipped models only) Button 7—the humidity sensor— is usually associated with variable VAR7.

1. Under **Point Mnemonic** enter the mnemonic of the variable that will store the humidity value.
2. Set **Type** to **Analog**.

3 Set **Range** to **0 Decimal**.

4 Set **Display** to **Read/Write**.

Operators can then view the room humidity by pressing Button 7 but cannot change the value.

Displaying time Typically button 5 is assigned to display time.

1 Under **Point Mnemonic**, choose the mnemonic of the variable that will store the time data. Typically this is variable VAR5.

2 Set **Type** to **Analog**.

3 Set **Range** to **Time** which will automatically format the display with a colon (:).

4 Add a Control Basic line as follows:

```
VAR5 = TIME
```

Verifying a functioning NetSensor To check if a functioning NetSensor is present, use the Control Basic function *NETSENSOR-STATUS* on page 300 .

```
IF NOT NETSENSOR-STATUS THEN STOP OUT1
```

Auxiliary function The auxiliary contact wires on the back of NetSensors with LED displays provide a method to indicate the state of an external switch or set of relay contacts.

1 When the wires are connected together, the **Aux.** value indicates **On**.

2 Under **Point Mnemonic**, enter the mnemonic of the variable that will store the **Aux.** value.

3 **Type**, **Range** and **Display** are preprogrammed for a digital input and cannot be changed.



Note: NetSensor models KMD–1161 and KMD–1181 do not have auxiliary wires. Simultaneously press buttons 5 and 7 and then press either the up arrow or down arrow button to change the auxiliary function from *Off* to *On*. NetSensor model KMD-1162 does not include an auxiliary function.

Motion sensing (Motion sensing models only) Detects movement in the room.

1. Under **Point Mnemonic**, choose the mnemonic of the variable that will store the state of Aux. Typically this is Variable VAR8 configured as an analog variable.
2. Set *Type* to **Analog**. **Range** is automatically set to **0 Decimal**.
3. Set **Display** to **Read/Write**.
4. Use Control Basic to test the state of the variable assigned to Aux.
   - 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 VAR9.

```
IF+ VAR8 = -1 THEN START VAR9 , STOP A
IF VAR8 = -2 THEN START A
IF TIME-ON( A ) > 0:02:00 THEN STOP VAR9
```

NetSensor button functions

The functions of the NetSensor buttons are listed in the following table.

Table 42–2 NetSensor button functions

Button	Function
Up arrow	Increases analog values and toggles digital values
Down arrow	Decreases displayed values; toggles digital 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 variable within the controller.
Buttons 3-6	Can be defined as any point available on the Tier 2 network.
Button 7	On applicable models, assigned as the humidity sensor physically included on the NetSensor.
Auxiliary	For use with auxiliary two-wire input on the back. Use with dry contacts such as a switch. NetSensors KMD–1161 and KMD–1181 do not have auxiliary wires. Press buttons 5 and 7 together and then press an up or down arrow button to change **Auxillary** from *Off* to *On.*

NetSensor tab properties

The properties in the NetSensor tab define how each of the buttons function.

Calibration Enter a calibration value that is added Button 1.

- 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° Fahrenheit above or below the displayed value.

Auto Blanking When this check box is selected, the NetSensor display will go dark approximately 15 seconds after the last button is pushed.

Point Mnemonic The mnemonic for the variable that stores the button value. The description can be a mnemonic from either the connected controller or any other controller on the Tier 2 network.

Value Displays the value, in engineering units, of the point listed in *Description*.

Type Sets the button as either an analog or digital function.

Range Select a unit of measure from the *Range* list.

- Units listed in the table *NetSensor analog display formats (continued)* on page 605 are active when *Type* is set to *Analog*
- Units listed in the table *NetSensor binary display formats* on page 605 are active when *Type* is set to *Digital*.

Display Read/write status of the button.

- *Read Only* indicates the operator may only view the data displayed on the NetSensor.
- *Read/Write* indicates an operator may use the buttons to change the data.

Table 42–3 NetSensor analog display formats

Range 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 variable.
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 variable.
Decimal_0	Sets the precision of the display to the nearest whole number.
Decimal_1	Sets the precision of the display to one place to the right of the decimal point.
Decimal_2	Sets the precision of display to two places to the right of the decimal point.
Time	Sets the NetSensor to display a time format.

Table 42–3 NetSensor analog display formats (continued)

Range unit	Action and display
RTC_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 42–4 NetSensor binary display formats

Range unit	Action and display
Off_On	The NetSensor toggles between words as arrow buttons are pressed and released. The digital point toggles between 0 and 1.
Low_High	The NetSensor toggles between words as arrow buttons are pressed and released. The digital point toggles between 0 and 1.
Cool_Heat	The NetSensor toggles between words as arrow buttons are pressed and released. The digital point toggles between 0 and 1.
Off_Low_High	The NetSensor cycles through each word as arrow buttons are pressed and released. The analog point cycles from 0 to 2.
Off_On1_On2_On3	The NetSensor cycles through each word as arrow buttons are pressed and released. The analog point cycles from 0 to 3.

Output points-analog

An analog output point sets the characteristics of an analog signal at the output of a controller.

General Properties

Name The 8-character description of the point. **Name** must be unique within the KMD controller that maintains it. The set of characters entered for **Name** must be printable characters.

Description A 20-character description of the device connected to the point. **Description** must be unique within the KMD controller that maintains it. The set of characters entered for **Description** must be printable characters.

Binary State When selected, the output is a binary output. When the Binary State check box is clear, the output is set as an analog output.

Value The current state of the point.

Units Choose the unit of measure and scale factor for the output signal. Outputs can be set up for one of several standard or custom units. For a list of units, see *Supported KMD controller engineering units* on page 621.

Manual Override Indicates the output is either in automatic or manual mode. When **Manual Override** is selected, the value will remain as displayed until changed by an operator. When the check box is clear, the value of the output signal can be set by either an operator or a calculation in a Control Basic program.

Switch If the controller is set to WinControl Compatible mode, **Switch** displays the state of the output override card. The position of the switch can be evaluated with Control Basic. See the keyword topic *OUTPUT-OVERRIDE* on page 308.

Table 42–5 Switch settings for KMD binary output points

Switch position on override card	State of switch
Hand	Manual
Off	Manual
Automatic	Auto

Security Level Choose an output security level to assign to the point. A user must have a security level at or above this value to make changes to the point.



Note: The Security Level property applies only when connecting to the controller with WinControl XL Plus. It does not apply to KMC Connect.

Delay Not available in analog input points.

Voltage Low Sets the voltage at the output which corresponds to 0% output. Applicable only when **Units** is set to Percent or Percent Open.

Voltage High Sets the voltage at the output which corresponds to 100% output. Applicable only when **Units** is set to Percent or Percent Open.

Alarm Routing Properties

Alarm classes designate a group of operators that receive notifications of alarms and events originating from this point. Select the KMC Connect alarm class routing from the list. See the topics *An overview of notifications, alarms, and events* on page 133, *Emailing operator alarms* on page 171, and *Alarm classes for KMD controllers* on page 153.

Related topics

- *Configuring KMD controllers and points* on page 581
- *Output points-binary* on page 607 to change an analog output to a binary output

Output points-binary

A binary output point sets the characteristics of an binary signal at the output of a controller.

General Properties

Name The 8-character description of the point. **Name** must be unique within the KMD controller that maintains it. The set of characters entered for **Name** must be printable characters.

Description A 20-character description of the device connected to the point. **Description** must be unique within the KMD controller that maintains it. The set of characters entered for **Description** must be printable characters.

Binary State When selected, the output is a binary output. When **Binary State** is clear, the output is set as an analog output.

Value The current state of the point.

Units Choose the unit of measure and scale factor for the output signal. Outputs can be set up for one of several standard or custom units. For a list of units, see *Supported KMD controller engineering units* on page 621.

The units of measure are shown as pairs. The first unit is the normal (not inverted) state. When referencing digital points in Control Basic, the left hand state is *False* (0) and the right hand state is *True* (1). Custom Units do not have an inverted range. If an inverted custom range is required you must reverse the setup on the custom unit or create a reversed custom unit.

Invert Polarity Sets the relationship between the physical state of the output and the logical state represented by **Value**.

Table 42–6 Output point polarity relationships

Invert Polarity	Voltage at output	Text displayed	Example
Clear	0	Off	Off On
Clear	10	On	Off On
Selected	0	On	Off On
Selected	10	Off	Off On

Manual Override Indicates the output is either in automatic or manual mode. When **Manual Override** is selected, the state of the output will remain as displayed until changed by an operator. When the check box is clear, the state of the output signal can be set by either an operator or a calculation in a Control Basic program.

Switch If the controller is set to WinControl compatible mode, **Switch** displays the state of the output override card. The position of the switch can be evaluated with Control Basic. See the keyword topic *OUTPUT-OVERRIDE* on page 308.

Table 42–7 Switch settings for KMD binary output points

Switch position on override card	State of switch
Hand	Manual
Off	Manual
Automatic	Auto

Security Level Choose an output security level to assign to the point. A user must have a security level at or above this value to make changes to the point.



Note: The Security Level property applies only when connecting to the controller with WinControl XL Plus. It does not apply to KMC Connect.

Delay Specifies a delay, in hours, minutes and seconds, which sets a period before the output changes to *ON* after being *OFF*. This timer takes precedence over any Control Basic program.

- Delay is entered in hours:minutes:seconds.
- The maximum start delay is 1:39:50.

Alarm Routing Properties

Alarm classes designate a group of operators that receive notifications of alarms and events originating from this point. Select the KMC Connect alarm class routing from the list. See the topics *An overview of notifications, alarms, and events* on page 133, *Emailing operator alarms* on page 171, and *Alarm classes for KMD controllers* on page 153.

Related topics

- *Configuring KMD controllers and points* on page 581
- *Output points-analog* on page 605 to change a binary output to an analog output

PID loop controllers

A PID controller point manages the PID loops in the connected controller. A PID loop is a mathematical function that calculates the analog output required to maintain a process at or near a setpoint.

See the related topic *Configuring KMD controllers and points* on page 581.

To use the output of a PID loop, assign the output of the PID loop controller to an output or variable point in a Control Basic program.

```
OUT1 = CON1
```

Controlled Variable Reference The mnemonic of the input measuring the condition under control. The controlled point is the sensed value that is to be maintained at the setpoint value.

Controlled Variable Value (Input) Displays the value of *Controlled Point*.

Controlled Variable Units Choose the unit of measure and scale factor for the controlled point signal. For a list of available units, see *Supported KMD controller engineering units* on page 621.

Manual Override Indicates the PID loop is either in automatic or manual mode. When **Manual Override** is selected, the value of the loop output will remain as displayed until changed by an operator. When the check box is clear, the value of the loop output signal is controlled by the PID algorithm.

Present Value (Output) Displays, in percent, the current output value of the controller. The value of the output is the mathematical sum of the four functions (PROP +RESET +RATE +BIAS).

Setpoint Reference Enter the target value of *Controlled Point*. Use a mnemonic, label or description to define *Setpoint*.

Setpoint Value Displays the value of the setpoint.

Action The action of the controller. This can either be direct-acting or reverse-acting.

For direct-acting loops, select **Positive**. Direct-acting controllers increase the output as the input rises above the setpoint.

For reverse-acting loops, select **Negative**. Reverse-acting controllers decrease the output as the input rises above the setpoint.

Proportional Constant The proportional constant is the value of the proportional gain parameter– the ratio of output response to the error signal–used by the loop algorithm. It represents the magnitude of sensed change that will cause the output to move from 0 to 100%.

For example, if the error signal has a magnitude of 10, setting the proportional constant to 5 would yield a proportional response of 50.

In general, increasing the proportional constant increases the response speed of the control system.

The proportional parameter is the *P* in PID.

To program *Proportional* with Control Basic, see *CONPROP* on page 264.

Integral Constant (Reset) Integral Constant (Reset) 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. *Reset* rate is specified in a repeats per hour (H) or per minute (M).

For example, a system maintaining a continuous error of two units and a reset of five per hour were specified, the output of the controller will change by ten units per hour in a direction to reduce the error.

Integral Constant (Reset) can also be thought of as the time integral of the error. Integral is the *I* in PID.

KMD PID controllers have an anti-windup feature when using *Integral Constant (Reset)*. In typical loops, the integral will continue to increase as long as there is a positive error. For most HVAC applications, such as space temperature control, integral windup is undesirable. To avoid windup, KMC PID controller algorithms are programmed with an integral capping feature. This limits the integral portion of the output to the percentage below 100% not used by proportional correction. If the proportional value is 60%, the integral can only increase to 40%. As the error increases, the rate that the integral adds into the output will increase but the integral will actually decrease because it has less room to operate. The proportional amount will continue to squeeze out integral until the input value is outside the proportional band, at which point integral value is 0. Integral windup can occur only within the proportional band and is eliminated as the error increases.

To program *Integral Constant (Reset)* with Control Basic, see the Control Basic keyword *CONRESET* on page 265.

Integral Constant Units Sets the time units for the *Integral Constant (Reset)* as either hours (the default) or minutes.

Derivative Constant (Rate) [in minutes] Rate slows the rate of change of the error. Rate is specified in minutes.

Use rate 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 .25 Min =15%. This 15% would be added in over the 1 minute in a direction to reduce the rate of change regardless of whether the input is above or below the setpoint.



Caution: Use *Derivative Constant (Rate) [in minutes]* 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.

Rate is the *D* (derivative) in PID.

Note that derivative is typically used only in systems without time lags and seldom is used in HVAC systems.

To program *Rate* with Control Basic, see the Control Basic keyword *CONRATE* on page 264.

Bias Sets the output value at setpoint. The bias is the value the controller will reach at equilibrium when derivative is not used.

To program *Bias* with Control Basic, see the Control Basic keyword *CONBIAS* on page 263.

For more infomation on PID loops, see Application Guide AG190614, PID Loops in KMC controllers.

Control Basic programs

A program point provides a network-visible view of the state of a Control Basic program within a KMD controller. It is also a method to open Control Basic programs that are edited with the KMC Connect code editor.

Name Short name of the program. Also referred to as the label.

Description Long name of the program

On Select to enable the program.

Manual When selected, the program cannot be executed by another Control Basic program.

Timer State (Tier 1 only) Check to run the program at the interval specified in *Repeat Minutes*.



Note: Do not select *Timer State* if the program includes Control Basic *NPAGE*, *TPAGE*, or *WAIT* statements.

Repeat Minutes (Tier 1 only) Sets the interval timer.

Remaining Minutes (Tier 1 only) Time left before program will execute.

Program Size (Tier 1 only) Amount of memory the program is using.

Exit Flag Control Basic forced an exit from the program area.

Related topics

- *Configuring KMD controllers and points* on page 581
- *About Control Basic programs* on page 217
- *Programming with the Code Editor* on page 195

Runtime Log points

Use Runtime Log points to record the number of cycles on a digital point and save the cumulative total of runtime hours for that point.



Note: Use a Runtime Log only to record events that are longer than one-second. Recording shorter events will result in missed or erroneous data.

General Properties

Trend Point Enter the mnemonic for the point to log. Points can be inputs, outputs, or variables. The point can be from the controller that contains the log or from other controllers on the network.

Total On Time Displays the total amount of time the point has been on since the start date. *On Time* is displayed in hours: minutes. Clear the data in the log by clicking **Reset Runtime Log** at the top of the tab.

Start Date The date the log began accumulating data.

Total Starts Displays the total number of starts since the log started.

Starts Today Displays the number of starts for the current calendar day.

Head Index The next location for storing data in the data buffer.

Wrap Flag This property is set to *True* when the data buffer is filled and newer data is writing over older data.

Reset Flag When this property is *True* an operator has sent a new configuration.

Log Buffer Size Displays the maximum number of samples that can be recorded in the runtime log. The buffer size depends on the type of controller.

- Tier 1 controllers—254 start/stop pairs
- Tier 2 controllers—150 start/stop pairs

Runtime logs and loss of power

All runtime data is stored in RAM memory in the KMDigital controllers. In Tier 2 controllers, RAM memory may not be backed up with battery power. This can lead to loss of data during a power failure. To guard against data loss, the total runtime hours and total number of cycles are automatically written into nonvolatile memory every day. When power is restored to the controller, the runtime totals are automatically carried forward as a starting point for the accumulation process.

Viewing Runtime Logs

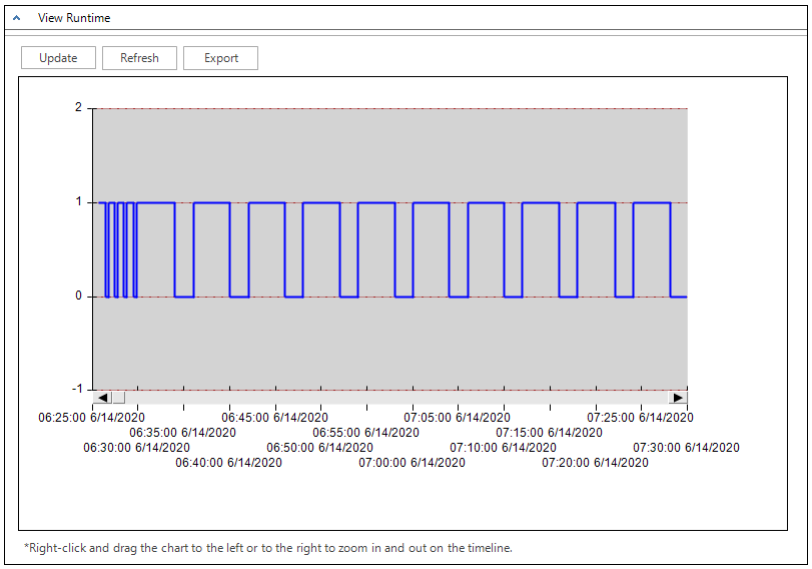
Choose **View Runtime** to display the Runtime Log data stored in a controller.

Update Click to retrieve data not on the screen and then add it to the screen data.

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

Export To export log data as comma separated values, click **Export**. Each data value is exported with a time stamp.

Illustration 42–3 KMD Run Time log data



Programming for Runtime Log

Control Basic can read total hours from a Runtime Log. The On Time is automatically rounded to whole hours.

Using a Runtime Log in Control Basic:

```
IF+ TIME > 9:00:00 AND DOW = TUE THEN GOTO 20 ELSE GOTO
30
IF RT1 > RT2 then START OUT2 ELSE START OUT1 : REM Line
20
END : REM Line 30
```

In this example pumps are controlled by outputs OUT1 and OUT2. The output OUT1 is logged in runtime log RT1, and the output OUT2 is logged in the runtime log RT2. The pump with the lowest runtime will be chosen to run at 9:00 AM each Tuesday.

```
ALARM RT1 > 299 , 1 , CHANGE AHU#1 FILTER. PRESS
PUSHBUTTON ON UNIT TO RESET TIME COUNTER
IF+ IN1 THEN START RT1
```

Starting a Runtime Log with the Control Basic command START resets the accumulated *On* time and total number of cycle counts to zero. This is equivalent to manually changing *Start Date* on a Runtime Log setup worksheet, except *Start Date* remains the same as before.

System Groups

Use Sytem Groups points to improve performance of TotalControl web pages with KMD points on a graphics page. Every time a point is added to a System Group, that point is added to the System Group list to be used by the KMD PDS and Web Portal. Add the

points for a page to a System Group and then bind the points to elements on the page. Points in a system group are then retrieved with a single network request rather than with a request for each individual point.

To add a point to a System Group, do the following:

1. In the Network Manager list, locate the network icon, Tier 1 device or Tier 2 device that contains the system group.
2. Locate and double-click the system group icon to open the configuration tab.
3. In the Network Manager list, locate a point to add to the system group.
4. Drag the point from the Network Manager list to the KMD Points list in the System Group tab.

Adding points from a graphics page

To add all of the points on a graphics page to a System Group, do the following:

1. Expand the Network Manager list until you locate the System Group icon to which you will add the points.
2. Drag a page from the Site Explorer list to the system group icon in the Network Manager list.

Reading the System Group list

The System Group list is automatically updated every time a change is made to a System Group from within KMC Connect. If a change to a system group from outside of KMC Connect—for example with WinControl XL Plus or KMC Connect—the list is not updated. Use the Update System Group List command to refresh the list at either the network or device level.

To read the system group list, do the following:

1. In the Network Manager list locate the network icon, Tier 1 device or Tier 2 device from which the system group list will be updated.
2. Right-click on the device or network icon and then choose **Read System Groups** from the shortcut menu.

Tables

A table performs a look-up function based on the actual input voltage of an analog input. The table converts the input voltage to a present value that is displayed in engineering units.

A table consists of up to 15 data pairs stored in two columns.

- Values in the X column correspond to an input voltage.
- Values in the Y column are the corresponding present values expressed in engineering units.

X Value The input voltage that will correspond to a specific present value in engineering units. Values in the X Values column must be entered in ascending order.

Y Value The present value–in engineering units–that corresponds to the X Value input voltage. The table function performs a linear interpolation between data pairs to calculate a Y value.

Units Select a unit of measure from the *Units* list. For a list of units, see *Supported KMD controller engineering units* on page 621.

Related topics

- *Configuring KMD controllers and points* on page 581
- The keyword *TBL* on page 329 for using tables in Control Basic

Trend Log points

Use Trend Log points to periodically collect and store data from key points in the system.

When the log becomes full, the oldest data is deleted and replaced with the latest readings. This creates a moving window of the most recent information.

Note: Use a trend log to record events that are longer than two or three seconds. Recording shorter events will result in missed or erroneous data.

For the procedures to set up Trend Logs, see the topics *Configuring KMD trend logs and groups* on page 121 and *Configuring KMD controllers and points* on page 581.

General Properties

Name The 8-character description of the Trend Log. Name must be unique within the KMD controller that maintains it. The set of characters entered for Name must be printable characters.

Log Interval Specify the time interval between readings. The interval is entered as *hours*:*minutes*:*seconds*. Enter a Log Interval value of zero (0) to log data under control of a Control Basic program.

Note: Once a value is entered in Log Interval, the Trend Log immediately changes to automatic sampling , all previous data is erased, and new data is recorded.

Trended Points Enter the mnemonic of the points to be recorded in the trend log. The logged points can be from within the controller or from other controllers on the network. The exact number of points placed in the list depends on the type of controller. See the topic *Mnemonics for KMD controllers* on page 239.

Control Basic Override (Tier 1 only) This check box is selected when the Trend Log is recording data on-demand instead of automatic sampling based on the period set by Log Interval. On-demand recording is under control of a Control Basic program. See *Configuring KMD controller Trend Logs* on page 121.

Trend Buffer Size Each log can store a limited number of records based on the type of controller.

Tier 1 255 samples
Tier 2 400 samples

View Trend

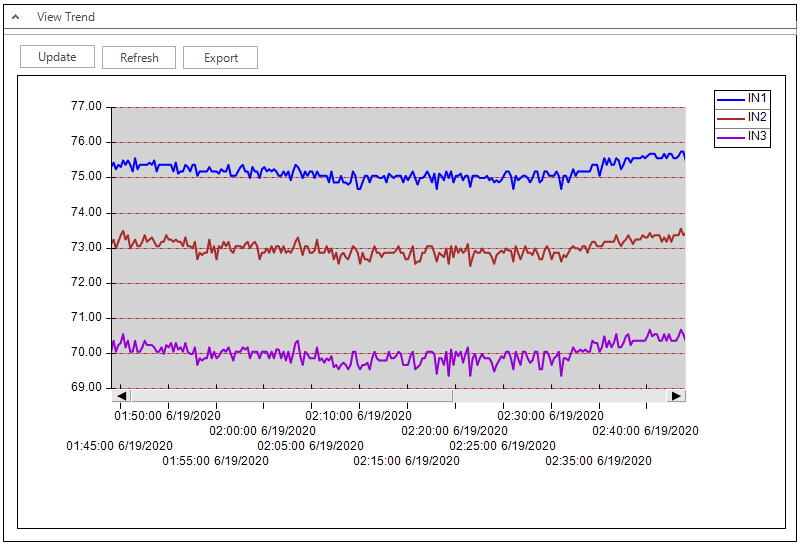
Choose **View Trend** to display the trend data stored in a controller.

Update Click to retrieve new data not on the screen and then add it to the screen data.

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

Export To export log data as comma separated values, click **Export**. Each data value is exported with a time stamp.

Illustration 42–4 KMD Trend point



Variable points-analog

Analog variable points are place holders for information such as setpoints temperature, time delays or other analog values. Use variable points as global program variables in Control Basic.

General Properties

Name The 8-character description of the point. **Name** must be unique within the KMD controller that maintains it. The set of characters entered for **Name** must be printable characters.

Description A 20-character description of the device connected to the point. **Description** must be unique within the KMD controller that maintains it. The set of characters entered for **Description** must be printable characters.

Binary State When the Binary State box is clear, the variable is an analog variable. When the Binary State check box is selected, the variable is set as a binary variable.

Value The current level, quantity, or state of the point.

Units Choose the unit of measure and scale factor for the input signal. See *Supported KMD controller engineering units* on page 621.

Manual Override Indicates the variable is either in automatic or manual mode. When **Manual Override** is selected, the value of the variable will remain as displayed until changed by an operator. When the check box is clear, the value of the variable can be set by either an operator or a calculation in a Control Basic program.

Alarm Routing Properties

Alarm classes designate a group of operators that receive notifications of alarms and events originating from this point. Select the KMC Connect alarm class routing from the list. See the topics *An overview of notifications, alarms, and events* on page 133, *Emailing operator alarms* on page 171, and *Alarm classes for KMD controllers* on page 153.

Related topics

- *Configuring KMD controllers and points* on page 581
- *Variable points-binary* on page 617

Variable points-binary

Binary variables are place holders for the state of binary data. Use variables as global program variables in Control Basic.

General Properties

Name The 8-character description of the point. **Name** must be unique within the KMD controller that maintains it. The set of characters entered for **Name** must be printable characters.

Description A 20-character description of the device connected to the point. **Description** must be unique within the KMD controller that maintains it. The set of characters entered for **Description** must be printable characters.

Binary State When selected, the variable is a binary variable. When the Binary State check box is clear, the variable is set as an analog variable.

Value The current state of the point.f

Units Choose the unit of measure and scale factor for the variable. See *Supported KMD controller engineering units* on page 621.

The units of measure are shown as a pairs. The first unit is the normal state. When referencing binary points in Control Basic, the left hand state is *False* (0) and the right hand state is *True* (1).

Manual Override Indicates the variable is either in automatic or manual mode. When **Manual Override** is selected, the state of the variable will remain as displayed until changed by an operator. When the check box is clear, the state of the variable can be set by either an operator or a calculation in a Control Basic program.

Alarm Routing Properties

Alarm classes designate a group of operators that receive notifications of alarms and events originating from this point. Select the KMC Connect alarm class routing from the list. See the topics *An overview of notifications, alarms, and events* on page 133, *Emailing operator alarms* on page 171, and *Alarm classes for KMD controllers* on page 153.

Related topics

- *Configuring KMD controllers and points* on page 581
- *Variable points-analog* on page 616

Weekly Schedule points

Use Weekly Schedule points to schedule recurring daily events. A scheduled time in a Weekly Schedule point sets the state of the point to *Active* when the time in the controller matches the scheduled time. Occupancy and lighting are two types of events that are typically scheduled with Weekly Schedule points. Weekly Schedule points include the following properties:

- Daily schedules for a seven-day week.
- Two override schedules for holidays or other special days.
- Both the daily and override schedules have several pairs of *On* and *Off* times that control the state of the schedule depending on the time of the day.

Once a schedule is entered, use Control Basic to test the state of the schedule to determine if it is *Active* (1, True, On, etc.) or *Inactive* (0, False, Off, etc.). To control a point with a weekly schedule, write a Control Basic line similar to the following example:

```
VAR1 = WS1
A = WS2
```

Annual Schedule points are usual points that control the overrides. However, a Variable or Input point can be used also to activate an override schedule. For example, a switch connected to an Input point or a Variable point can be used to activate one of the override daily schedules.

General Properties

Name The 8-character description of the point. **Name** must be unique within the KMD controller that maintains it. The set of characters entered for **Name** must be printable characters.

Description A 20-character description of the point. **Description** must be unique within the KMD controller that maintains it. The set of characters entered for **Description** must be printable characters.

Manual Select the check box to put the schedule in manual mode; clear the check box to put the schedule in automatic mode.

Present Value Indicates whether the schedule is *Active (On)* or *Inactive (Off)*.

Override Object 1 and Override Object 2 Enter the mnemonic of the point that will enable the override. See the topic, *Mnemonics for KMD controllers* on page 239.

- If an override is *True*, the state of the weekly schedule uses the scheduled times in the Override1 or Override2 columns.
- If Override Object 1 is *True* then Override Object 2 is disregarded.
- Override days are typically controlled by Annual Schedule points. See the topic *Annual Schedule points* on page 589.

Rules for Weekly Schedule points

Follow these rules when entering Weekly Schedule values.

- Enter times using the 24-hour clock format.
- Entering 00:00 clears the field.
- When more than one *On* and *Off* time is entered for the same day, times must be entered in ascending order.
- Enter 00:01 (1 minute after midnight) to designate midnight.
- Do not leave empty days. If the *On* or *Off* state is to continue from a previous day, enter 00:01 in the first *On* or *Off* row for that day.
- Override 1 and Override 2 schedule times are used when the value controlled by the point entered in the Override 1 or Override 2 column becomes *True* (1) or *On*.

Alarm Routing Properties

Alarm classes designate a group of operators that receive notifications of alarms and events originating from this point. Select the KMC Connect alarm class routing from the list. See the topics *An overview of notifications, alarms, and events* on page 133, *Emailing operator alarms* on page 171, and *Alarm classes for KMD controllers* on page 153.

Related topics

- *Configuring KMD Weekly Schedule points* on page 190
- *Configuring KMD Annual Schedule points* on page 191
- *Annual Schedule points* on page 589
- *About KMD Schedules* on page 189

Section 43: Supported KMD controller engineering units

KMC Connect supports the engineering units for KMD controllers listed in this section.

Table 43–1 Units for KMD analog inputs

Amperes 0 to 100	Percent 0 to 100 OR	Table1
C KMC10k Type II	mA 4 to 20	Table2
C KMC10k Type III	Percent 0 to 100 OR	Table3
Counts 2x32	V 0 to 5	Table4
F KMC10k Type II	Pulses Per Min 0 to	Table5
F KMC10k Type III	255	Table6
FPM 0 to 3000	CCMH	Unused
mA 0 to 20 Input	counts	V 0 to 5

Table 43–2 Units for KMD analog outputs

mA 0 to 20
Percent
Percent Open
PSI 0 to 20
V 0 to 10

Table 43–3 Units for KMD analog outputs

Amperes	Custom7	Minutes
BTU	Custom8	Ohms
C	Days	Pa
CF	F	Percent
CFH	FPM	Percent Open
CFM	GPM	Percent RH
Counts	Hours	PSI
Custom1	In W	Pulses per min
Custom2	KPa	Seconds
Custom3	KVolts	Time
Custom4	KWatts	Unused
Custom5	KWH	Volts
Custom6	mA	Watts

Table 43–4 Units for KMD digital inputs, outputs and variables

Closed Open	Custom6	Normal Alarm
Cool Heat	Custom7	Normal High
Custom1	Custom8	Normal Low
Custom2	Custom_digital8	On Off
Custom3	Disabled Enabled	Stop Start
Custom4	Low High	Unoccupied
Custom5	No Yes	Occupied

Section 44: Reference to KMD controllers

This appendix list total of each type of point for each KMC controller family.

The information in the following table lists the number of points in KMD 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 of a point, see *Reference to KMD points* on page 589.

Notes:

- Model KMD–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 44–1 Tier 1 controllers

Type	5205	5210	5270
Alarms	192	192	192
Annual Schedule	4	16	4
Arrays	8	48	8
Control Basic	10	128	10
Inputs	8	128	8
Outputs	8	128	8
Passwords	256	256	256
PID Control loops	8	64	8
Runtime Logs	16	128	16
Sign On log	32	32	32
System groups	32	64	32
Tables	5	5	5
Trend Logs	16	96	16
Variables	128	256	128
Weekly Schedules	8	32	8

Table 44–2 Tier 2 controllers

Type	5801	5821	5831	6000	7000	7300
Alarms	10	10	10	1	10	10
Annual Schedule	2	2	4	-	0	0
Arrays	n/a	n/a	n/a	n/a	n/a	n/a
Control Basic	5	5	10	4/1	5	5
Inputs	8	8	12	4	4	4
Outputs	8	8	16	4	4	4
Passwords	27	27	27	2	27	27
PID Control loops	8	8	16	4	4	4
Runtime Logs	8	8	12	2	2	2
Sign On log	n/a	n/a	n/a	n/a	n/a	n/a
System groups	4	4	8	1/1	2	2
Tables	3	3	6	2/1	3	3
Trend Logs	8	8	12	2	2	2
Variables	64	64	128	32	32	32
Weekly Schedules	4	4	8	1	1	1

Section 45: KMD Tier 1 alarm messages

KMD Tier 1 controllers issue alarm notifications when an internal error is detected. This section is a list of the error messages.

Table 45–1 Tier 1 alarm messages

Module	Alarm Text	Cause	Restorable
Cbasic	Program Error: Prog	WS-ON: WS-OFF: An illegal routine# FOR TO NEXT: An off-panel point is used as value to increment INC: sub-opcode other than RD_A-RD_Z, or RD_PNT DEC: sub-opcode other than RD_A-RD_Z, or RD_PNT	No
Cbasic	Off Panel Write : Prog #, Line # , OpCode #!	STOP, DISABLE, CLOSE, LET: Argument is an off panel point.	No
Cbasic	Invalid Table: Prog #, Line # , OpCode #!	TBL: Invalid table#.	No
Cbasic	Array Table Invalid: Prog #, Line # , OpCode #!	Invalid array index used anywhere an array point is read or is assigned a value.	No
Cbasic	Invalid OpCode: Prog #, Line # , OpCode #!	Can be tested by using an email command on a LAN Controller	No
Cbasic	BACnet Not Licensed: Prog #	These errors are not currently implemented in Control Basic.	No
Cbasic	BACnet Application Layer Not Configured: Prog #	These errors are not currently implemented in Control Basic.	No
Cbasic	BACnet: Ethernet is not Running: Prog #	These errors are not currently implemented in Control Basic.	No
Cbasic	Main Panel Does not Exist: Prog #, Line # , OpCode #!	STATUS: Illegal panel#	No
Cbasic	User Name not found: Prog #, Line # , OpCode #!	PRINT USER-A/B: There is no user logged-on through port A/B.	No
Cbasic	LSEL too many values: Prog #, Line # , OpCode #!	LSEL: The number of expressions is less than two or greater than 30.	No

Table 45–1 Tier 1 alarm messages (continued)

Module	Alarm Text	Cause	Restorable
Cbasic	HSEL too many values: Prog #, Line # , OpCode #!	HSEL: The number of expressions is less than two or greater than 30.	No
Cbasic	Probable Infinite Loop: Prog #, Line # , OpCode #!	More than 25,000 OpCodes have been executed for one pass of the program.	No
BACnet	BACnet Service Request Timeout on Device #, Object #	A BACnet read or write property request has timed out.	No
BACnet	BACnet Who-Is Timeout on Device #	A BACnet who-is request has timed out.	No
BACnet	BACnet MS/TP Sole Master	No other panels are accepting the token.	Yes
MainNet	Too Many MainNet In Want Points! Point not added:	There are more than 64 In Want Points.	No
MainNet	Too Many MainNet Out Want Points! Point not added:	There are more than 64 In Want Points.	No
MainNet	MAIN PANEL IS OFFLINE-#	Five minutes have passed since this panel was heard from.	Yes
SubNetA	Too Many SubNetA In Want Points! Point not added:	There are more than 512 in Want Points.	No
SubNetA	Too Many SubNet A Out Want Points! Point not added:	There are more than 64 In Want Points.	No
SubNetA	SUB-PANEL IS OFFLINE M#-SUBA-#	Five minutes have passed since this panel was heard from.	Yes
SubNetB	Too Many SubNetB In Want Points! Point not added:	There are more than 512 In Want Points.	No
SubNetB	Too Many SubNet B Out Want Points! Point not added:	There are more than 64 In Want Points.	No
SubNetB	SUB-PANEL IS OFFLINE-SUBB-#	Five minutes have passed since this panel was heard from.	Yes
Ethernet	Trying to send more packets than allowed by protocol!	User response message length greater than 255 bytes. Broadcast message length greater than 1500 bytes.	No
Ethernet	Panel to Panel message from unknown panel : # port #	MSG_TOPANELS received from a panel not in the Netlist.	No

Table 45–1 Tier 1 alarm messages (continued)

Module	Alarm Text	Cause	Restorable
Ethernet	Broadcast message from unknown panel : # port #	MSG_TOBEHOST received by a panel that is not a broadcast server. MSG_TOBEHOST received from a panel not in the Netlist.	No
Ethernet	PANEL IS OFFLINE-#	Five minutes have passed since this panel was heard from.	Yes
Hourly Memory Check	Memory utilization has exceeded 95%! Memory utilization has exceeded 90%! Memory utilization has exceeded 85%! Memory utilization has exceeded 80%! Memory utilization has exceeded 75%!	This could display after loading the panel with programs, schedules, and system groups. A memory leak would also be detected by these alarms after running the panel for an extended period of time.	No

KMC Connect

Part IX: Working with OPC objects

Section 46: An introduction to OPC

The topics in this section are an overview of the OPC process.

OPC is a non-proprietary standard that specifies the communication of real-time data between control devices from different manufacturers. It is a standard that defines a set of interfaces, based upon Microsoft's OLE/COM technology, that is often used in industrial automation and enterprise systems. The application of the OPC standard interface makes possible interoperability between building automation, manufacturing control applications, field systems, devices, business, and office applications.

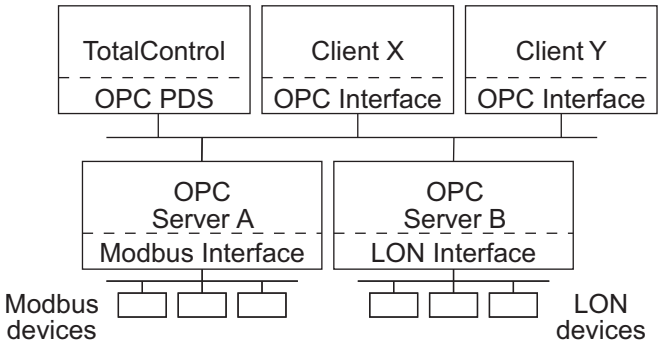
OPC is the original name for *OLE for Process Control*, an open standards specification developed in 1996 by an industrial automation industry task force.

OPC has two parts: a client and a server.

Client A client program connects to and obtains data from an OPC server. The OPC client can be on the same computer or on another computer. The client makes a request to the server for data and the server then fulfills the request. For a KMC Connect managed site, the OPC PDS is the client application that connects the OPC sever to the components of KMC Connect.

Server An OPC Sever is a software application that acts as a protocol converter. The OPC server connects to devices or systems and translates the data into a standard OPC format. Each supported protocol will have a server application written specifically to obtain data from a source in the protocol and then convert it to the OPC standard format. OPC client applications— the KMC Connect PDS for example—can then connect to the OPC server and use it to read and write device data.

Illustration 46–1 Typical OPC network with KMC Connect



Section 47: Setting up an OPC service

The OPC service is the link between a third-party OPC server and other KMC Connect services. The following topics are a reference for setting parameters in the OPC service.

The OPC service is distributed as part of KMC Connect. Typically, it is installed when KMC Connect is installed.

Topics covered in this section:

- *Configuring an OPC service* on page 634
- *Connecting to an OPC server* on page 633

Related topics:

- *An introduction to OPC* on page 631
- *Reference to OPC objects and tags* on page 637

Connecting to an OPC server

The KMC Connect OPC service can connect to either a local or remote OPC server.

- A local OPC server runs on the same computer on which the KMC Connect OPC service is running.
- A remote OPC server runs on a computer separate from the computer on which the KMC Connect OPC service runs. The computer on which the OPC server runs must also have DCOM enabled or is running a DCOM tunneling program.

Connect to a local OPC server

1. In the Network Manager list, right-click the OPC Service icon .
2. In the shortcut menu choose **Find/Regenerate Objects**.
3. The OPC service will connect to the local server and then add OPC groups and items to the Network Manager list.

Connect to a remote OPC server

1. In the Network Manager list, right-click the OPC Service icon .
2. In the shortcut menu choose **Find/Regenerate Remote Objects**. The New OPC Server URL dialog opens.
3. Enter the name of the computer on which the OPC server is running.
4. The PDS will connect to the remote server and then add OPC groups and items to the Network Manager list.

Configuring an OPC service

Use the OPC Service configuration tab to view or change parameters of the OPC service.

1. Locate the OPC Service icon in the Network Manager list.
2. Right-click the OPC 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** in the tab of the configuration page.

Set OPC Service as Default

The default service in TotalControl can be set to either BACnet, KMD, or OPC. TotalControl graphic elements with a service ID of 0 will use this service.

To set the OPC Service as the default:

1. Locate the OPC Service icon in the Network Manager list.
2. Right-click the OPC Service icon and choose **Set as Default** from the shortcut menu.



Note: "Set as Default" is unavailable if the service is already the default.

3. When changes are complete, click **Save Changes to Service** in the tab of the configuration page.

General Properties

Items under General Properties are usually set up when the OPC service was added to the Network Manager list.

Service Name A descriptive name of the service to display in the Network Manager list.

Service ID The Service ID number that was automatically assigned by KMC Connect at the time the service was added to the site database.

Service Host Address This is the address of the computer on which the SQL service is running. The preferred entry is the name of the computer; the IP address may also be used.

Service Host TCP Port This port is used by KMC Connect building services. For KMD networks the default TCP Port is 27031.

Settings and Statistics

Maximum Response Staleness Sets the age of the data. If—at the time data is requested from an OPC server—the data in the database is older than the value of *Maximum Response Staleness*, KMC Connect will retrieve new data from the server. If the data is newer than the value in *Maximum Response Staleness*, KMC Connect displays the value stored in the database.

Monitor Statistics Select to enable the service to save statistical data for diagnostics.

Incoming Requests The number of requests the service has received from KMC Connect. During device discovery, this will peak and then subside as discovery is completed.

Pending Requests The number of unanswered requests that the PDS has sent to the networks. During discovery the rise and fall of this value will somewhat lag the incoming requests rise and fall.

Completed Requests The number of requests sent to networks for which the service received a response. This is a cumulative value since the service was last started. Selecting the *Monitor Statistics* check box will periodically initiate requests of the service, which increases the *Completed Requests* by *one* each time a request is made.

Related topics

- *Connecting to an OPC server* on page 633
- *Adding a service* on page 44
- *Opening and closing the Network Manager pane* on page 42

Section 48: Reference to OPC objects and tags

Topics in this section are a reference to the features and characteristics of OPC objects.

OPC objects in the Network Manager list represent the groups, items and tags as provided by the OPC server to which the OPC PDS is connected.

- To connect a KMC Connect site to an OPC server, see *Connecting to an OPC server* on page 633.
- To locate an OPC object on an OPC server that is connected to a KMC Connect managed site, see *Using the Network Manager* on page 41.
- For an overview of OPC, see *An introduction to OPC* on page 631.

Object Name A description of the OPC tag.

Present Value The current level, quantity, or state of the OPC tag.

Data Type Present Value can be any of the following data types. Data types define a set of values and the allowable operations on those values. OPC data types are listed in the table *Standard OPC data types (continued)*.

Table 48–1 Standard OPC data types

Description
Default/Empty (nothing)
2-byte signed integer
4-byte signed integer
4-byte (single-precision) real
8-byte (double-precision) real
Currency
Date
Text (UNICODE)
Error code
Boolean (TRUE = -1, FALSE = 0)
1-byte signed integer
1-byte unsigned integer
2-byte unsigned integer
4-byte unsigned integer

Table 48–1 Standard OPC data types (continued)

Description
Array of 2-byte signed integers
Array of 4-byte signed integer
Array of 4-byte (single-precision) real
Array of 8-byte (double-precision) real
Array of currency values
Array of dates
Array of text values
Array of error codes
Array of Boolean values
Array of 1-byte signed integers
Array of 1-byte unsigned integers
Array of 2-byte unsigned integers
Array of 4-byte unsigned integers

Timestamp The date and time of the most recent reading of Present Value. Timestamp is typically the time of the Present Value reading in the device. However, if the device the does not provide a time stamp, it is added by the OPC server to which the PDS is connected.

Quality An indication of the quality state of the tag's present value. Quality is based on a major quality value, a substatus for that major quality value, and a limit status indicating how the value is limited.

Table 48–2 Supported OPC quality flags

Flag	Description
NotSet	The quality is not set.
Good	
GoodLocalOverride	Overridden; usually, input disconnected and manual value forced.
Bad	Bad, but no reason available
BadLastKnown	Communication failed - last known value
BadCommFailure	Communication failed - no last known value
BadWaitingOnInit	No error - default for items added to subscription until read

Table 48–2 Supported OPC quality flags (continued)

Flag	Description
BadConfigError	There is a server-specific problem with the configuration.
BadInputDisconnected	The input appears to be disconnected.
BadDeviceFailure	Device failure detected
BadSensorFailure	Sensor failure detected
BadOutOfService	Block is off scan or locked
Uncertain	Uncertain with no reason available
UncertainLastUsable	Providing last usable value
UncertainSensorCalibration	Sensor is pegged at a limit or out of calibration
UncertainEUExceeded	Value outside defined limits
UncertainSubnormal	Value derived from less than required number of sources

KMC Connect

Part X: Protocol Gateway Service and Protocol Gateway Manager

Section 49: Overview of the Protocol Gateway

The Protocol Gateway is a KMC Connect service that transfers values between controllers of different protocols. This section is an overview of Protocol Gateway Manager and the Protocol Gateway service.

Two components of KMC Connect are required to transfer values between dissimilar protocols.

- The Protocol Gateway service is a component of KMC Connect.
- The Protocol Gateway Manager is a programing tool within KMC Connect.

Protocol Gateway service The KMC Connect Protocol Gateway service monitors data in one or more points or objects and then transfers that value to another point or object. The transfer takes place at regular intervals that range from every few seconds to once a month. For example, the present value of a BACnet object can be transferred to a variable point in a KMD controller.

The Protocol Gateway service runs only as long as KMC Connect is running on the computer.

Protocol Gateway Manager The Protocol Gateway Manager is the tool that technicians use to set up a transfer process for points or objects between controllers of different protocols. It also sets an interval between transfers.

Related topics *Setting up Protocol Gateway processes* on page 645.

Section 50: Setting up Protocol Gateway processes

Topics in this section cover setting up Protocol Gateway processes with the Protocol Gateway Manager.

The Protocol Gateway service transfers values between different types of KMC Connect supported BAS protocols with one or more defined processes. The processes are set up with the Protocol Gateway Manager in KMC Connect.

- A process input transfers values to a process output.
- The input can be a user defined value or a point or property in the Network Manager list.
- The process output is a point or property in the Network Manager list.
- A process runs at designated intervals that can range from a few seconds to monthly.

Related topics

- *Using the Protocol Gateway Manager* on page 645
- *Managing Protocol Gateway processes* on page 647
- *Overview of the Protocol Gateway* on page 643

Using the Protocol Gateway Manager

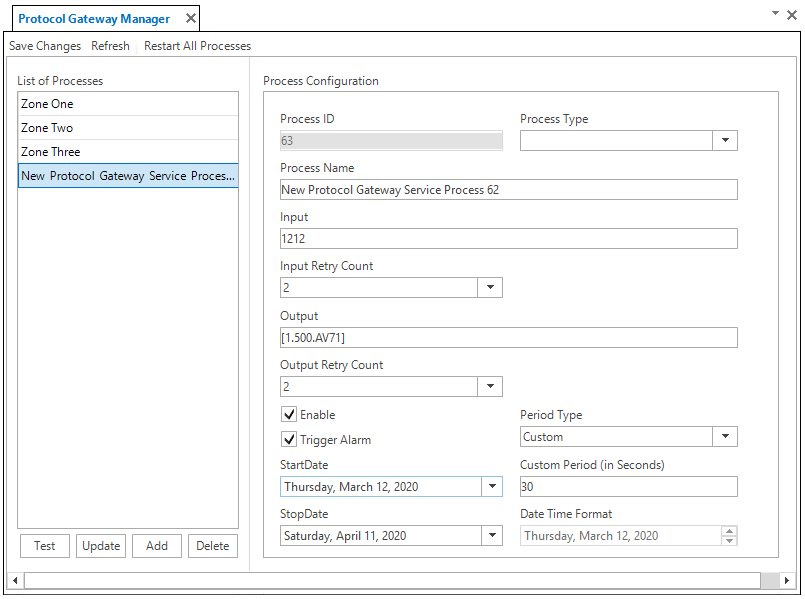
The Protocol Gateway Manager is the KMC Connect tool that technicians use to set up a gateway process.

To open the Protocol Gateway Manager, do the following:

1. In the Network Manager list, locate the Protocol Gateway service icon .
2. Right-click the icon and then choose **Protocol Gateway Manager** from the shortcut menu. The Protocol Gateway Manager tab opens.
3. Add, delete, test, or restart processes as described in the topic *Managing Protocol Gateway processes* on page 647.

4 At the top of the tab, click **Save Changes** when all settings are complete.

Illustration 50–1 Protocol Gateway Manager



Toolbar commands

Save Changes Saves the changes to all processes in the site database.

Refresh For the selected process, the parameters in the Protocol Gateway Manager are discarded and the parameters from the site database are loaded.

Restart Restarts all processes that have stopped.

The components of Protocol Gateway Manager

Test The selected process runs once and then stops until the next interval. The interval for transferring a value is specified in **Period Type**.

Update Stores the selected process in the site database.

Add Places a new process in the List of Processes.

Delete Removes a process.

Process ID A unique number assigned by the Protocol Gateway service. It cannot be changed.

Process Name A unique label that identifies the process. Use only letters, numbers and the underscore (_) symbol for the process name.

Input Assigns a point or property in the Network Manager list, or a fixed-value to the process input.

Input Retry Count Sets the number of times the Protocol Gateway service will attempt to read the value of the input process.

Output Assigns either a point or property in the Network Manager list as the process output.

Output Retry Count Sets the number of times the Protocol Gateway service will attempt to change the value of the point or property assigned to the process output .

Enable Select the check box to enable the process; clear the check box to disable the process.

Trigger Alarm When selected, a message will be placed in the Output Window list if the Protocol Gateway service cannot read a process input or change a process output.

Start Date and Stop Date Sets the effective period in which the process will run.

Process Type The process type designates the type of input for a process.

- **Integration Point** sets the process input as a point or property in the building automation system.
- **Fixed Value** sets the process to a user-defined value or constant entered in **Input**.

Period Type Sets the interval for running the process.

- Custom—Sets the interval to the value in **Custom Period**.
- Every Noon—The process runs once a day at noon.
- Daily—The process runs at the time specified in **Date Time Format**.
- Weekly—The process runs on the day of the week and at the time specified in **Date Time Format**.
- Every Two Weeks—The process runs every two weeks on the first day of the week and at the time specified in **Date Time Format**.
- Monthly—The process runs monthly on the day of the month and at the time specified in **Date Time Format**.
- One Time—The process runs once at the time in **Date Time Format**.

Custom Period Sets the interval—in seconds—at which a process will run.

Date Time Format Sets the time and day or date on which a process will run. The date and time for the process schedule is controlled by the clock in the computer on which the gateway service is running.

Related topics

- *Overview of the Protocol Gateway* on page 643
- *Managing Protocol Gateway processes* on page 647

Managing Protocol Gateway processes

A process transfers the value of a point or property from one controller to another. Usually, the controllers are of different protocols such as KMDigital and BACnet. A process can also write a fixed value to a point or property. The transfer takes place at designated intervals.

See the topic *Using the Protocol Gateway Manager* on page 645 for details on the Protocol Gateway Manager tab.

Create a new process

1 Open the Protocol Gateway Manager.

2 Do one of the following:

- Click **Add**.
- Right-click in the **List of Processes** area and then select **Add Process** from the shortcut menu.

3 In **Process Name**, enter a unique name for the process. Use only letters, numbers and the underscore (_) symbol for the process name.

4 Designate an input for the process by doing one of the following:

- In the **Process Type** list, choose **Integration Point** and then drag a point or property from the Network Manager list to **Input**.
- In the **Process Type** list click **Fixed Value** and then enter a fixed value in **Input**.

5 Drag a point or property from the Network Manager list to **Output**.

6 Select the **Enable** check box.

7 Set the following interval properties:

- **Period Type**
- **Custom Period, Start Date, Stop Date** and **Date Time Format** as required by **Period Type**.

8 When all settings are complete, click **Save Changes** at the top of the tab.

Delete a process

1 Open the Protocol Gateway Manager.

2 Locate the process to delete in **List of Processes**.

3 Do one of the following:

- Select the process and then click **Delete**.
- Right-click the process and then choose **Delete Process** from the shortcut menu.

4 Choose **Yes** to delete and **No** to cancel.

Restarting processes

This procedure restarts all processes that have stopped.

1 Open the Protocol Gateway Manager.

2 At the top of Protocol Gateway Manager configuration tab, click **Restart All Processes**.

Testing a process

When a process is tested, it runs once and then stops until the next interval as specified by **Period Type**.

1. Open the Protocol Gateway Manager.
2. In **List of Processes**, select the process to test.
3. Click **Test**.

Section 51: Configuring the Protocol Gateway service

The KMC Connect Protocol Gateway service transfers values between different types of KMC Connect supported BAS protocols with one or more defined gateway processes.

To run gateway process from a KMC Connect managed building automation system, a Protocol Gateway service must be running on the computer that is running KMC Connect. Typically the Protocol Gateway service is installed at the same time that KMC Connect is installed.

To open the Protocol Gateway service, do the following:

1. In the Network Manager, locate the Protocol Gateway service icon .
2. Right-click the Protocol Gateway 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** in the tab of the configuration page.

Properties of the Protocol Gateway service

Service Name A descriptive label of the service. Service Name is the name that identifies the service in the Network Manager list. The set of characters used in Service Name is restricted to printable characters.

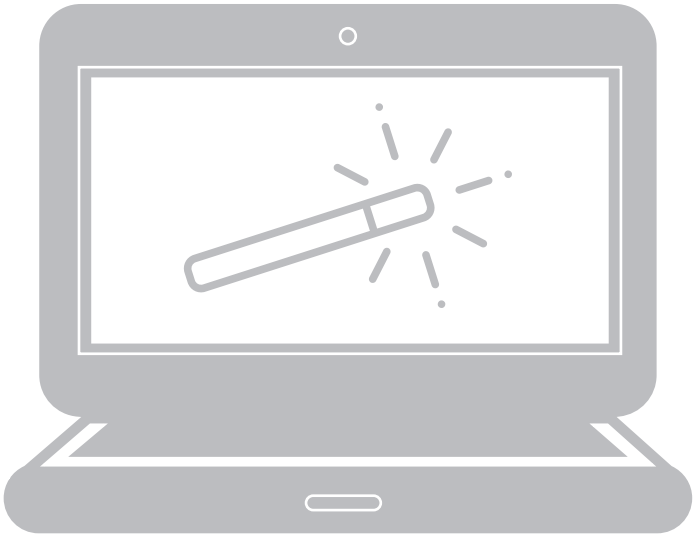
Service ID Service ID is assigned by KMC Connect and cannot be changed.

Service Host Address Not applicable for KMC Connect.

Service Host TCP Port Not applicable for KMC Connect.

KMC Connect

Part XI: Applications and wizards



Section 52: Application selection, configuration, and deployment

The Selection, Configuration, and Deployment tools with which a technician can configure controllers from the KMC Controls applications library.

Use the Selection, Configuration, and Deployment applications to manage the options in programs supplied by KMC Controls in *KMC Conquest-compliant controllers* on page 709. 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 controller application. The settings are saved as templates in the selected application job.

Deployment For deployment, templates are matched to, and then written to, individual controllers.

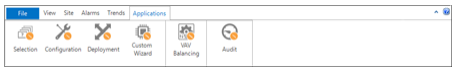
Details of the applications.

- 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.

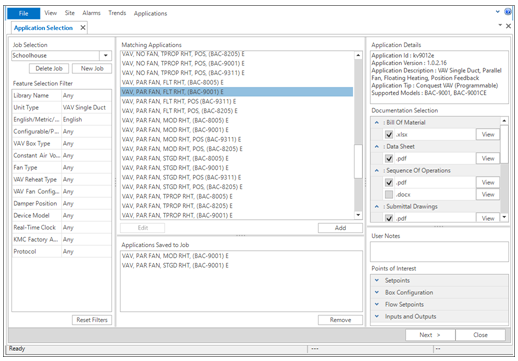
Step 1—Selection

The Selection page sets up application jobs.

1 On the ribbon, choose **Selection** from the **Applications** group.



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.

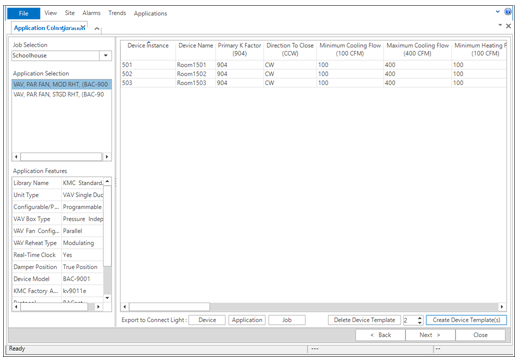
5 Under **Matching Applications**, select the model or models that are the best fit for the application. Click **Add** to save them to the job. More than one type of application can be added to each job.

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.

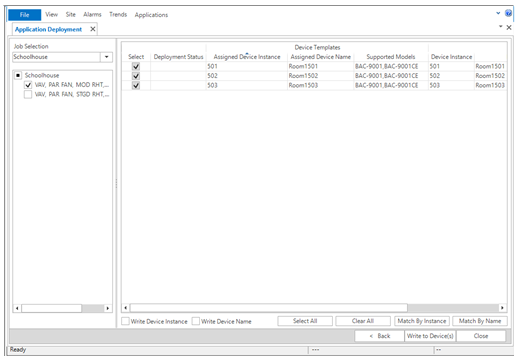
4 When finished, click **Next**. The templates are stored in the job file and the application advances to Deployment.

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 53: Audit application

The Audit application is a tool to detect changes to the configuration of controllers on the building automation system.

Use the Audit application to establish a baseline of performance and then, in the future, compare current settings to the stored baseline.

Using the Audit application is a two part process.

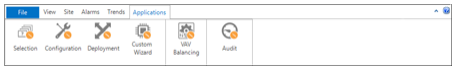
- After controllers are set up and commissioned, the values of the points-of-interest are saved as a baseline.
- To verify performance, baseline values are compared to current values.

For a list of controllers compatible with the application, see *KMC Conquest-compliant controllers* on page 709.

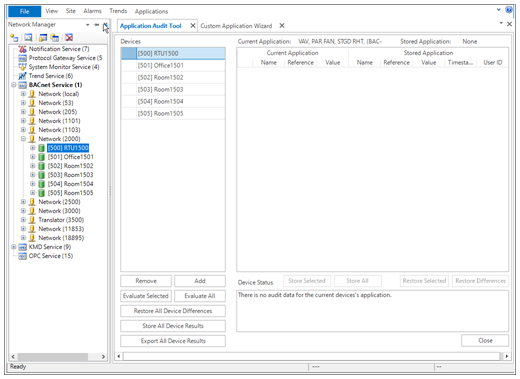
Establishing the baseline

Before establishing an audit baseline, set up the controllers. Once the baseline is established, it can be used at any time to detect changes to the system.

1 On the ribbon, choose **Audit** from the **Applications** group.



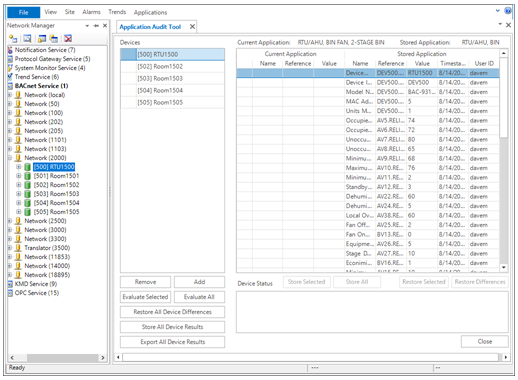
2 Open the Network Manager list and drag a device, network with devices, or the BACnet service icon to the Devices area. Only devices that are Conquest compliant are available for audit. All others are marked with a yellow triangle ⚠.



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.



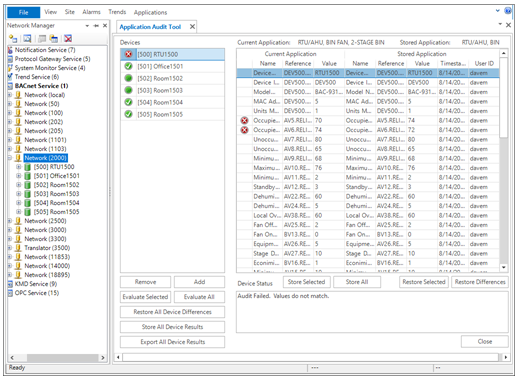
4 Save the baseline by clicking **Store All** or **Store Selected**. This adds the values to a stored baseline for future reference.

Comparing controllers to a baseline

For controllers with an established baseline, the current values in the controllers can be evaluated at any time against the baseline values.

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 54: 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 includes the following features for balancing, configuring, and commanding airflow in compatible VAV controllers.

- *The VAV Balancing sequence* on page 663 is a tool that control technicians can use to balance the airflow in a VAV unit.
- *VAV flow monitoring* on page 666 can be used to monitor airflow through a VAV unit without starting the balancing sequence in a controller.
- *Commanding VAV airflow* on page 666 temporarily sets a specific airflow without changing the permanent VAV properties.

For a list of controllers compatible with the VAV Balancing application, see the topic *KMC Conquest-compliant controllers* on page 709.

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 *KMC Conquest-compliant controllers* on page 709.



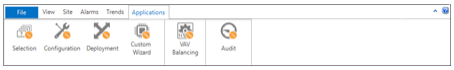
Note: The procedures in this section are for pressure independent systems only.



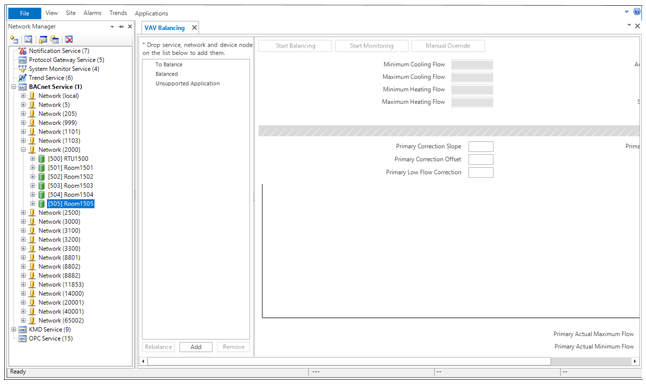
Tip: Once balancing has started, all steps must be finished in order. If the procedure is interrupted, unpredictable VAV operation will result.

To run the VAV Balancing sequence, do the following:

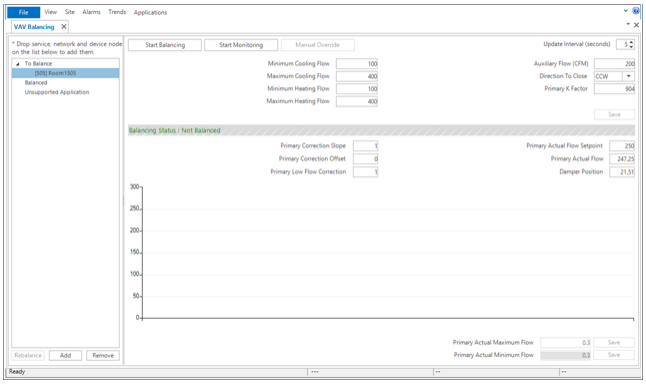
1 On the ribbon, choose **VAV Balancing** from the **Applications** group.



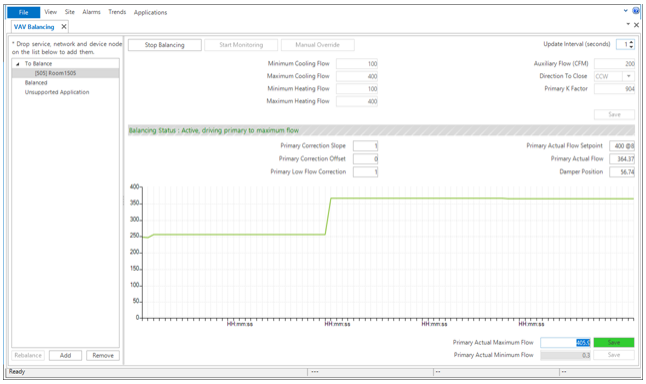
2 Open the Network Manager list and drag a device, a network with devices, or the BACnet service icon to the **To Balance** area.



3 Select an individual device and then click **Start Balancing**. The application commands the VAV controller to position the damper for maximum airflow.



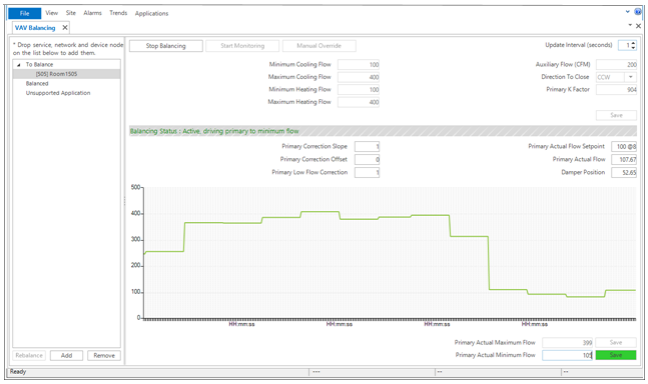
4 Monitor the airflow graph and the value in Primary Actual Airflow until the airflow stabilizes at or near the maximum airflow value.



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 starts moving 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 airflow values are displayed by the application. Both airflow values 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.

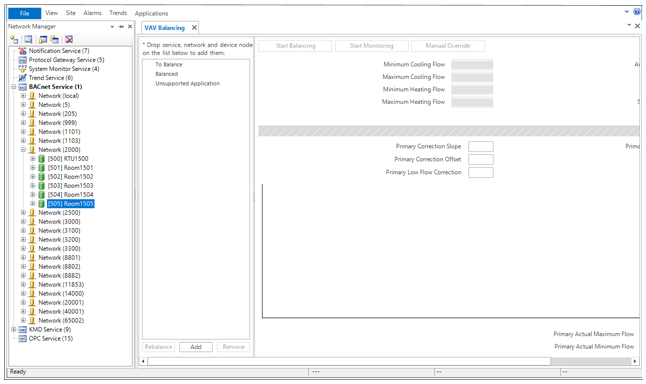
To balance the airflow for a VAV unit, see the topic *The VAV Balancing sequence* on page 663.

To monitor airflow through a VAV unit, do the following:

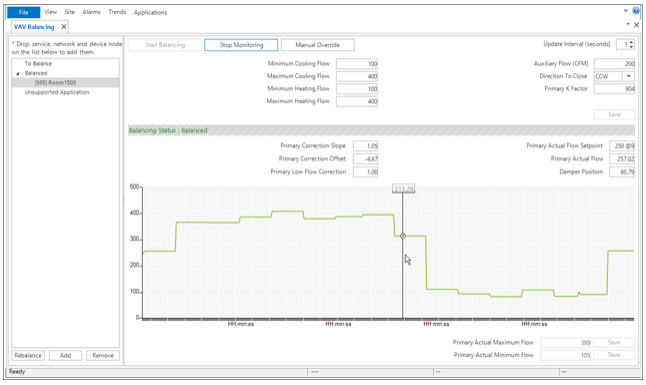
1. On the ribbon, choose **VAV Balancing** from the **Applications** group.



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.



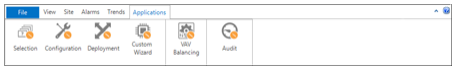
4. When finished, click **Stop Monitoring**.

Commanding VAV airflow

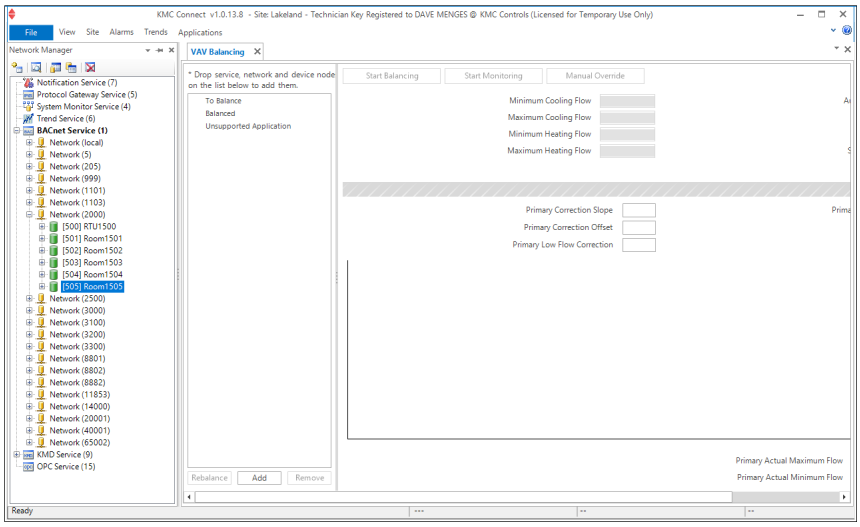
Use the VAV airflow manual override function to command airflow to a specific airflow value. When a controller is in manual override, the calculated airflow setpoint is replaced with a temporary fixed setpoint. This function is used to make adjustments or measurements that require a steady, specific airflow.

To command a specific airflow through a VAV unit, do the following:

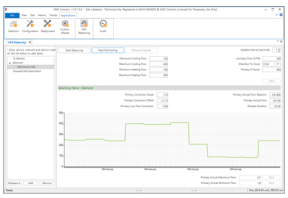
1 On the ribbon, choose **VAV Balancing** from the **Applications** group.



2 Open the Network Manager list and drag a device, a network with devices, or the BACnet service icon to the **To Balance** area.

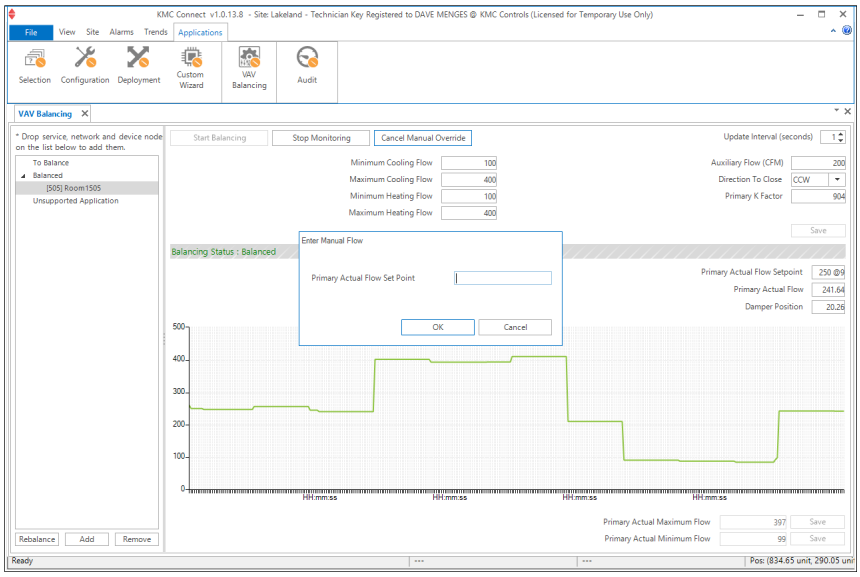


3 Select an individual device and then click **Start Monitoring**.



4 Click **Manual Override**.

5 In the **Enter Manual Airflow** dialog, enter an airflow setpoint and then click **OK**.



6 When finished, click **Cancel Manual Override**.

Section 55: 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 Applications Custom Wizard to save a custom application based on the configuration in a controller that contains a standard KMC application. The application can then be used with the Selection, Configure and Deployment applications. It can also be used to define points-of-interest for the Application Audit Tool for older KMC BACnet controllers. Custom applications are saved in the Resource Manager 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 that is configured in a controller.
- Define a custom set of points-of-interest that can be used with the Audit Applications Tool.
- Add documents relevant to the application.
- Save all modifications, object assignments, Control Basic programs, sensor configurations, etc. in a `.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.

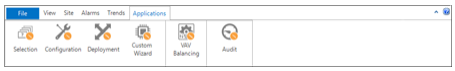
Note: Before starting the wizard, complete the configuration of the controller that will be the template for the custom application.

Setting up the application

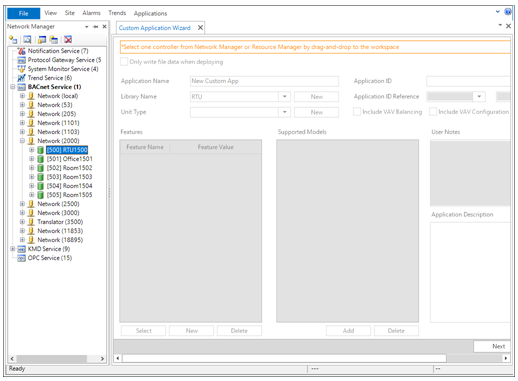
This procedure starts building a custom application. It sets up the application name, features, and supported models. If needed, it also sets up a new custom application library in the Resource Manager.

To start the custom application, do the following:

1 On the ribbon, choose **Custom Wizard** from the **Applications** group.



2 Open the Network Manager list and drag the controller with the custom application to the orange workspace area.



3 Select **Only write file data when deploying**. This sets up the `.bnd` file for a BACnet write-file procedure when used with the Deployment application.

4 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 application. This will also be the name of the folder in the Custom Applications library.
- **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.
- 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 Controls or custom name.
- **Application ID**—Enter a unique identification for the application that is stored as a property in the Device object.
- **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.

5 Optional for VAV applications: Check the **Include VAV Balancing** and **Include VAV Configuration** check boxes if these features are required.

6 Optional: Click **Add** to add selection criteria when the application is used with the Selection tool.

- The feature name will appear in the Feature Selection Filter list in the Selection tool.
- The value is used to select a specific feature.

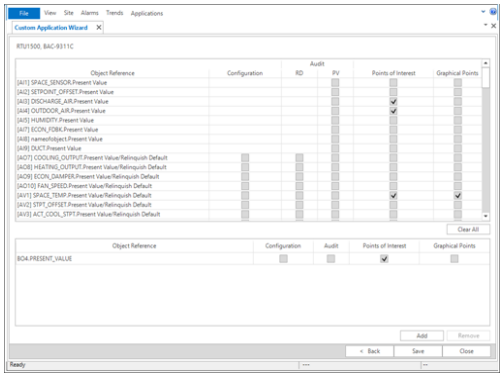
7 Select, add, or delete a supported model. The model of the template device cannot be deleted.

8 To add documents to the custom application library, select the **Include Documents** check box. Document selection will take place after the Points selection step.

9 When ready, click **Next** to advance to Points selection procedure.

Points selection

This procedure selects points that will be used by the Selection and Audit tools.



1 Select points to include in the custom application.

- **Configuration**—These are the 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.

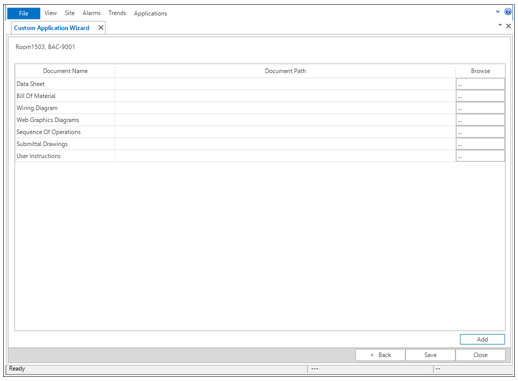
3 Click **Next** to advance to the document selection procedure.

- 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 now select documents to add to the custom application. In this procedure, selected documents are copied from their original location and then stored with the job in the Custom Affliction Library folder in the Resource Manager.

1. For each document type that will have a document added, click the ellipsis (...) under **Browse**.



2. When the Windows dialog opens, browse to the document to include in the application.
3. Optional: Click **Add** to create a new document category and then add documents to the category.
4. When all documents are added, click **Save** to save the custom application in the Custom Affliction Library folder in the Resource Manager.

Section 56: Input and output object wizards

Topics in this section cover the BACnet input and output object wizards.

Use the BACnet input and output wizards to configure the most common properties in input and output objects.

Each of the wizards are explained in one of the following topics.

- *Analog input object wizard* on page 673
- *Binary input object wizard* on page 676
- *Analog output object wizard* on page 677

Analog input object wizard

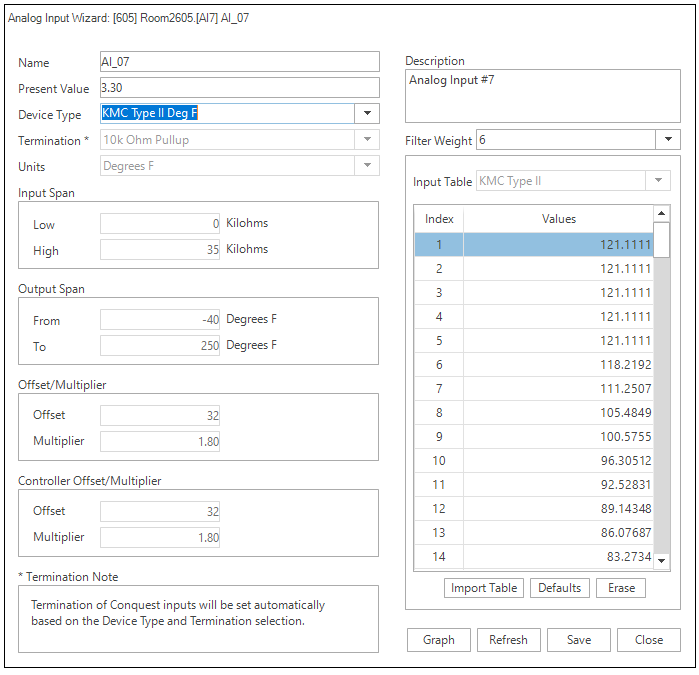
Use the Analog Input Wizard to match the BACnet Analog Input object and Input device table properties to the input device connected to the controller. These properties are fully described in the topics *Input objects-analog* on page 461 and *Input tables* on page 561.

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 56–1 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 physical 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 **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 500–2,000 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 Conquestseries 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 561 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 **Filter Weight**.

9 Click **Save** when finished.

Passive Custom device types

For passive input devices (thermistors, potentiometers,, 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 nonlinearity 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 561 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 676
- *Input objects-analog* on page 461
- *Input tables* on page 561

Binary input object wizard

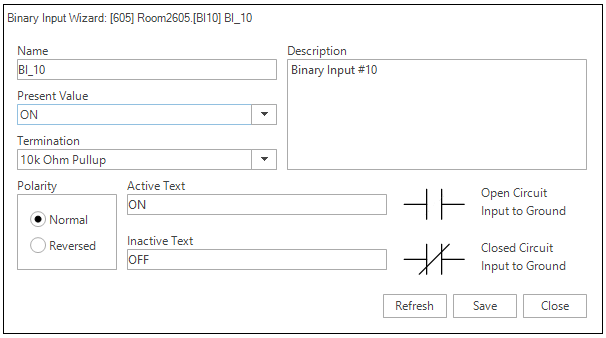
Use the Binary Input wizard to match the BACnet Binary Input object properties to the input device connected to the controller. These properties are fully described in the topic *Input objects-binary* on page 466.

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 56–2 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 interaction is shown in the following table.

Table 56–1 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 is automatically 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* on page 676.

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 673 and *Input objects-binary* on page 466.

Analog output object wizard

Use the Analog Output Wizard to match the BACnet Analog Output object to the output device connected to the controller. Properties for the object are fully described in the topic *Output objects-analog* on page 491.

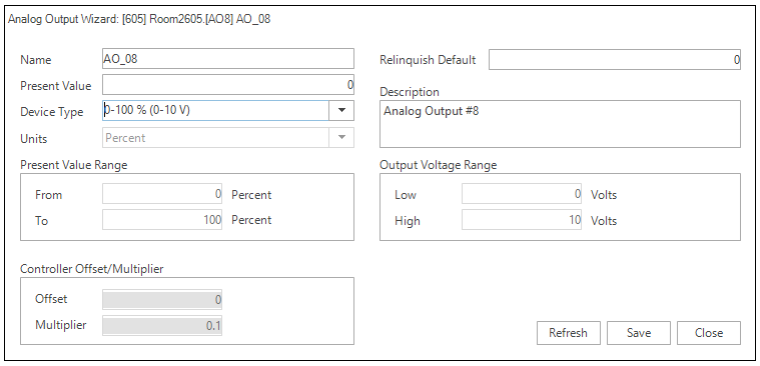
To open the Analog Output Wizard, do the following:

1 In the Network Manager list, locate the controller and then open the Outputs folder.

2 Locate an analog output icon .

3 Right-click the icon and choose **Analog Output Wizard** from the shortcut menu.

Illustration 56–3 BACnet Analog output wizard



The properties in the wizard are those properties required to match the signal from the output of the controller to the input of the device connected to it.

Standard device types

Choosing an item from the Device Type list automatically configures most of the output object properties.

To configure a standard device type, do the following:

1. From the **Device Type** list, select the entry that matches the type of device connected to the controller.
2. Select **Units** to match the type of output signal.
3. Enter a priority for writing in **Relinquish Default**.
4. Enter a name and description for the output.
5. Click **Save** when finished.

Custom device types

For output devices 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 connected to the controller. This information is either a voltage or current range.

1. From the **Device Type** list, choose **Custom**.
2. In **Output Voltage Range**, enter the **Low** and **High** range of the signal as specified by the device connected to the controller.
3. In **Present Value Range**, enter the **Low** and **High** values that will correspond to the Low and High values in Output Voltage Range.
4. Select **Units** to match the type of output signal.
5. Enter a priority for writing in **Relinquish Default**.
6. Enter a name and description for the output signal.
7. Click **Save** when finished.



Appendices

Appendix A: Communication ports and BAS networks

This topic focuses on Information Technology issues related to your Building Automation System (BAS).

A Building Automation System (BAS) consists of a network of hardware devices that communicate using one or more KMC Connect supported protocols. Software applications provide an interface between the user and these devices for monitor and control of the building environment. The devices and software perform most functions without manual intervention. They also collect trends of monitored sensor and equipment status, and will send alarm messages as needed.

Topics in the remainder of this section cover the following information:

- *Firewalls and ports* on page 681
- *Remote access* on page 683
- *BACnet controllers and networks* on page 684
- *KMD controllers and networks* on page 685

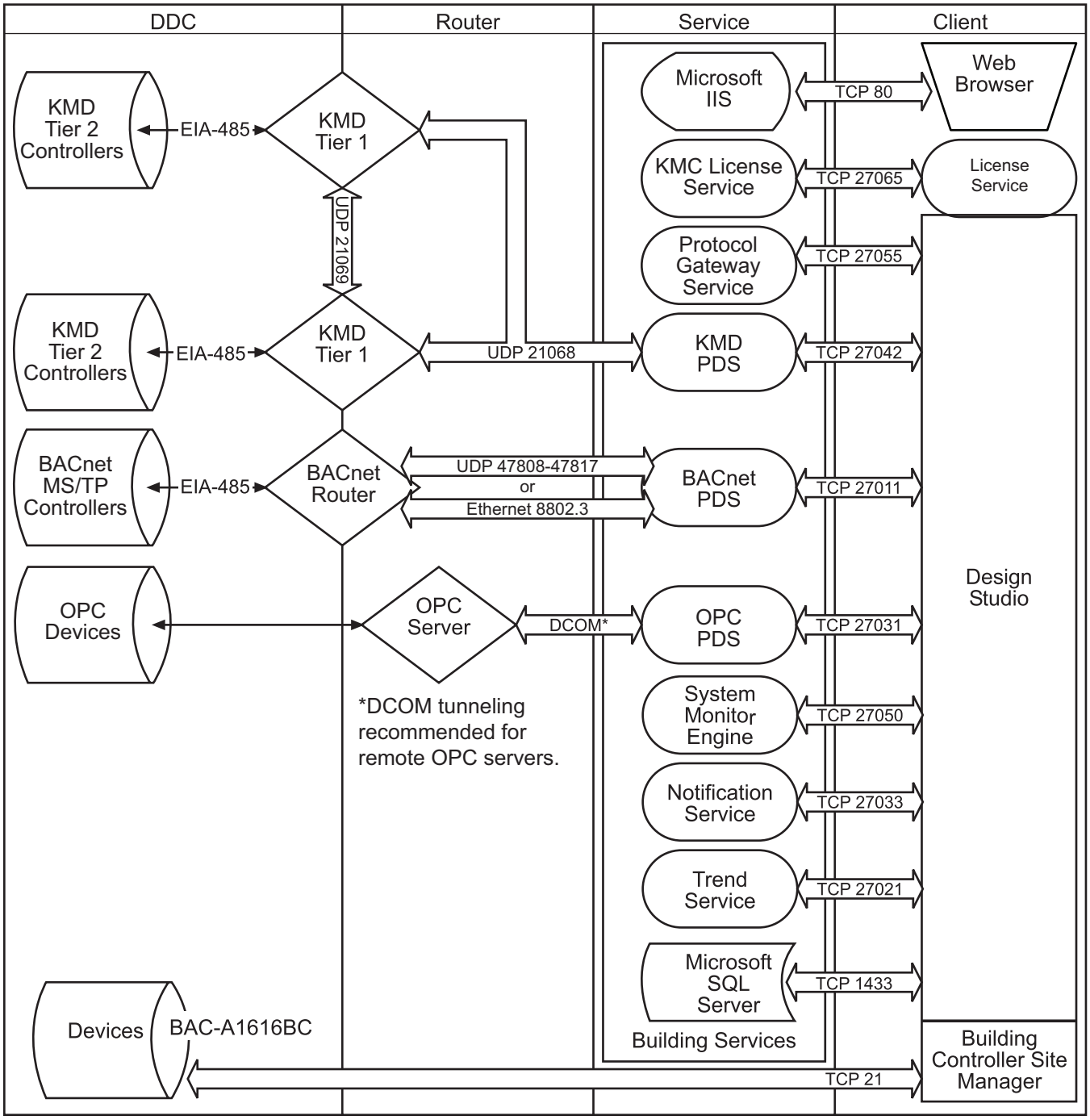
Firewalls and ports

Because a Microsoft Windows or Microsoft Server operating system has an inherent firewall, exceptions must be added to the firewall to allow communication through the firewall ports. The KMC Connect installer adds exceptions to the Windows firewall for its own services. Other ports, such as those for IIS, SQL and BACnet traffic, must be manually added to the firewall. If other firewalls—such as with network IP routers— are in the path between elements, exceptions for the ports must also be added to the firewall. Client-server communications between Building Services are not shown in the following table and diagram.

Table A–1 TCP/UDP Ports

Device / Application	Purpose	Port	Location
KMC License service	Active licensed components	TCP 27065	Anywhere
Building Controller Site Manager	Technician configuration tool	TCP 21	Anywhere
KMD Protocol Driver Service (PDS)	KMD BAS server	TCP 27042	Site
KMD Tier 1 controllers	KMD traffic router	UDP 21068	Site
KMD Tier 1 controllers	KMD traffic router between Tier 1 controllers	UDP 21069	Site
OPC Protocol Driver Service (PDS)	OPC server	TCP 27031	Site
BACnet Protocol Driver Service (PDS)	BACnet BAS server	TCP 27011	Site
BACnet routers	BACnet traffic router	Typical range is 47808-47812	Site
BAC-5050 & BAC-1616	Router configuration	TCP 21070	Site
System Monitor Engine	BAS server	TCP 27050	Site
Notification service	BAS server	TCP 27033	Site
Trend Service	BAS server	TCP 27021	Site
Protocol Gateway service	BAS server	TCP 27055	Site
MS IIS Web Server		TCP 80	Site
MS SQL Dbase server		TCP 1433, 1434	Site
Design Studio	Client (Technician configuration tool)	Communicates with services, IIS & SQL	Anywhere
Web browser	Client (End User)	Communicates with IIS	Anywhere

Illustration A–1 Communication ports for KMC Connect



Remote access

If remote access to the building automation system is required, we recommend setting up a VPN account for your authorized installer. This provides a secure, remote access for KMC Connect. We also recommend that remote connectivity—either by Microsoft's remote console or Virtual Network Computing (VNC)—be set up temporarily on the computer until installation of KMC Connect and communication with devices has been verified. This will enable technical service personnel to identify problems.

BACnet controllers and networks

The KMC Controls BACnet products in a Building Automation System (BAS) conform to the BACnet communications standard (www.bacnet.org) for monitor and control functions, services, devices, and software applications. The issues discussed are:

- Broadcast domains and traffic tunneling schemes in BACnet
- Network traffic loading

BACnet controllers are installed throughout the building to control building equipment and monitor environmental and system status. The controllers from KMC Controls use BACnet MS/TP (EIA-485 physical layer) for communication. Gateways, called BACnet routers in the standard, transfer traffic along EIA-485 and Ethernet cabling between DDCs and the BACnet Protocol Driver Service.

BACnet networks, broadcast domains and tunneling

The BACnet standard specifies networking schemes that may be used at OSI Layers 1-4. Several physical layers are supported, mainly EIA-485 and IEEE 8802.3. For IEEE 8802.3, BACnet traffic is transported either at the Link layer, called “BACnet over Ethernet”, or at the Network/Transport layers, called “BACnet over IP”, as follows:

- BACnet/Ethernet uses an LSAP of 0x82 in the LLC sub layer to identify BACnet traffic.
- BACnet/IP uses UDP ports to identify BACnet service traffic The default port is 47808 (0xBAC0).

Every device within the system has peer-to-peer communications with all other devices, routed through BACnet routers. BACnet networks are defined as follows:

- An Ethernet segment
- An IP broadcast domain, defined by a UDP port
- An EIA-485 network
- A one-to-one unicast tunnel between two routers, defined by a UDP port. This is referred to as the PAD method.
- A many-to-one unicast tunnel between routers and a single router configured as a broadcast manager, defined by a UDP port. This is the BACnet Broadcast Management Device (BBMD) and foreign device method.

BBMDs and Packet Assembler and Disassembler (PAD) routers are used to cross IT subnetwork boundaries. PAD routers carry broadcast and unicast messages between devices. In the BBMD method, only broadcast messages are tunneled between the many foreign devices and the BBMD. The BBMD informs all of its registered foreign devices of the entire set of foreign devices so that foreign devices will route unicast messages directly to other foreign devices. Thus, if a BBMD fails, this will stop the flow of broadcast—and eventually all—BACnet traffic within the BBMD's registered foreign devices.

The following are some defining features of BACnet routers:

- BACnet routers are really gateways.
- BACnet routers do not route Layer 3 (IP) traffic; they route BACnet traffic at all appropriate layers.
- BBMD traffic passing through a firewall to the Internet requires a public IP address. Network address translation is supported for BACnet routers that comply with Addendum O to BACnet Standard 135-2010 or Annex J of BACnet Standard 135-2012.

The Cimetrics BACstac driver—part of the BACnet Protocol Driver Service for KMC Connect—may be configured to route traffic as a either a BBMD or foreign device. Typically, a BBMD is used for permanent installations. A foreign device connection is only recommended for a temporary connection such as a technician's service tool.

Bandwidth and traffic loading

BACnet traffic mainly consists of periodic requests for sensor and equipment status. The periodicity can be minutes, even hours, and messages are very small. A site of 100,000 sq. ft. may typically have system-wide traffic on the LAN on the order of a few tens of packets a second to a few kilobytes a second.

Web traffic per client is also minimal. Typically, only a few hundred kilobytes are necessary to present a page in the Web Portal and to initialize a Flash movie or HTML 5 animation, and then a few kilobytes every 30 seconds thereafter.

Other applications and KMC Connect on a single computer

For a system in which BACnet traffic is transported over BACnet/IP, BACstage and KMC Connect may not operate through the same Ethernet network interface card (NIC) while using the same UDP port. To simultaneously use other applications, each application must be assigned its own network and UDP port; a matching UDP port and network must be configured on the internetwork within a BACnet router.

KMD controllers and networks

Only the Tier 1 KMD series[1] of controllers are directly connected to the Ethernet LAN. Tier 2 controllers are connected to Tier 1 controllers with EIA-485 networking. All controllers use points of data (such as an input from a sensor) to perform their control function(s).

[1]KMDigital is a proprietary line of digital building automation controllers manufactured by KMC Controls, Inc. 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 an EIA-485 network. KMDigital is often abbreviated as KMD.

Bandwidth issues

In general terms, KMD controllers take only a minutely small portion of available Ethernet bandwidth. Some bandwidth usage is fixed, other usage is dynamic, and still other usage is user-defined.

Fixed bandwidth Each controller is set to send and receive data only every 20 to 30 seconds. This traffic probably accounts for no more than 6 kilobytes at peak while at most times it is idle.

Dynamic Dynamic usage refers to alarm conditions. These exceptions occur rarely, even in large and complex automation systems. In addition, each alarm will account for only up to several hundred bytes or so of bandwidth.

User defined User-defined traffic is more unpredictable. Users can set up KMC Connect to access controllers and perform any number of tasks. Again, even at peak, this traffic accounts for only a small percentage of available Ethernet traffic bandwidth.

While not meant to be definitive or comprehensive, the following formula can provide a general calculation for controller-to-controller bandwidth.

max. bandwidth per controller =

3000 bytes X (number of Tier 1 controllers in system – 1) every 25 seconds

This formula does not account for dynamic or user-defined traffic.

Summary of KMD network use

- A Network Status UDP packet—UDP datagrams of 40 bytes—is sent every 23 seconds to each Ethernet connected controller in the system.
- A Network Points packet—a UDP datagram from 0 to 528 bytes—is sent every 24 seconds to each Ethernet connected controller in the system.
- Alarms—UDP datagrams of up to 1400 bytes—are sent to each Ethernet connected controller as they occur. Multiple alarms will be concatenated into one message. These same packets are then directed to the KMD PDS or operator workstation that is both logged in and set up to receive such alarms.
- Operator requests—UDP datagrams of 24 bytes—are directed to specific controllers. Replies can be up to 1400 bytes.
- Operator sends can be a UDP datagram of up to 1400 bytes; controllers acknowledge with a 15-byte UDP datagram.
- The building operator determines and controls the frequency of activity. Operator workstation refresh rates can be set as often as once per second, In addition, multiple windows may be open simultaneously. Only those windows that are in Update mode (as opposed to View mode), generate network traffic. However, workstations can also be programmed to maintain historical files known as trend logs. Workstation settings determine the number of trend logs and the frequency of updates. Trend log packets (UDP datagrams) are 1400 bytes.
- Packets transmitted on the same Ethernet sub-network are fragmented at the Maximum Transmission Unit (MTU) size set at the controller level not related to the

LAN MTU size. Default size is 1500 bytes.

- Routers and gateways have a tendency to drop KMD controllers from routing tables. To keep the controllers in the tables, Tier 1 controllers ping routers and gateways every 2 minutes.

Network protocols supported

- Internet Protocol (IP) over Ethernet
- Internet Control Message Protocol (ICMP)
- Universal Datagram Protocol (UDP)
- Address Resolution Protocol (ARP)
- BACnet-licensed KMD models (MS/TP, Ethernet 8802.3)
- Proprietary protocol (encapsulated in UDP)

Broadcasting on KMD networks

- In the traditional sense of the word, KMC controllers do not perform any broadcasting functions.
- Address Resolution Protocol (ARP) messages are used whenever a given Tier 1 controller needs to determine the MAC address of other controllers. This broadcast ARP is of no consequence to the other devices on the Ethernet LAN.
- We generally recommend that Tier 1 controllers are designated as broadcast servers in our facility management software. However, this traffic exists only from controller to controller.
- All UDP transmissions are directed (no broadcasts).

Appendix B: 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.

The Cimetrics BACstac driver is an automatic part of installing KMC Connect. The BACstac driver is a software service that connects the local BACnet network to the KMC Connect BACnet service.

- Prerequisite system information and requirements are covered in the topic *Before you begin* on page 689.
- Configuring the driver for the correct BACnet protocol
  - *Configuring BACstac for BACnet IP* on page 690.
  - *Configure BACstac for Ethernet 8802.3* on page 694.
- If needed, removal procedures are covered in the topic *Remove previous versions of the BACstac driver* on page 695.

Before you begin

The Cimetrics BACstac driver is included in the installation of 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. 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 match the BACnet protocol of the local BACnet network.

UDP Port The UDP Port number is assigned by a BACnet router or a device with a routing function. Each BACnet IP network must be assigned a UDP Port. The port number in BACstac must match the port number set by the router.

Related topics

- *Configuring BACstac for BACnet IP* on page 690
- *Configure BACstac for Ethernet 8802.3* on page 694
- *Remove previous versions of the BACstac driver* on page 695

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

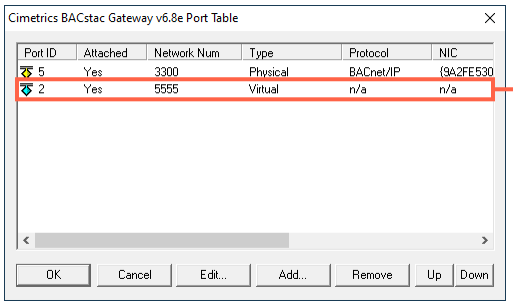
To configure the computer as a BACnet IP device, the computer must meet one of the following conditions:

- It must be connected to the same IP subnetwork as one of the BACnet IP networks.
- It must be configured as a BACnet Broadcast Management Device (BBMD) that is part of the BACnet internetwork. This is the recommended practice if no other means to send and receive BACnet broadcast messages is available.
- It must be registered as a foreign device to a BBMD. This is not recommend for permanent installations.

All three types of IP routing protocols require 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.

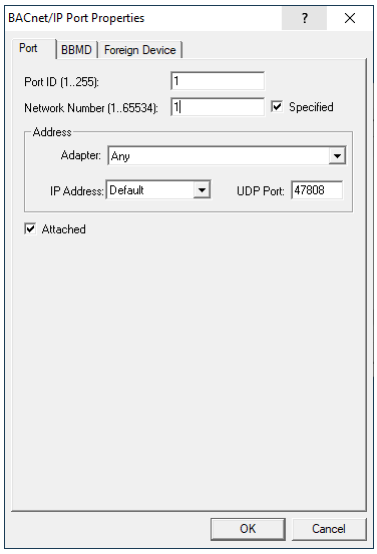
2 If there are any unneeded entries in the port table, select them and then click **Remove**.



3 Click **Add** and then choose **BACnet/IP**.

4 Enter the configuration for the following in the BACnet/IP dialog:

- Port ID
- Network number
- UDP Port



5 From **Adapter,** select the physical network adapter that will connect to a LAN.

6 If required, perform either of the following additional configurations:

- *Configure BACstac as a BBMD* on page 693
- *Register as a foreign device* on page 692

7 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 subnetwork 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

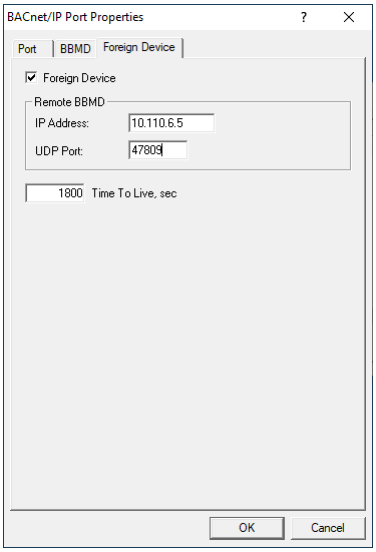
The computer can register as a foreign device to a BACnet Broadcast Management device (BBMD). Foreign device registration to a BBMD is a technique for crossing IP subnetwork boundaries with BACnet broadcast messages.

Note: Registering as a foreign device is not recommended for permanent connections. Instead, configure BACstac as a BBMD.

To configure the KMC Connect computer as a foreign device, do the following:

1 *Configure the computer as a BACnet IP device* on page 690

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 removes 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

A BACnet Broadcast Management Device (BBMD) facilitates BACnet broadcast messages across IP subnetworks. When configuring the driver as a BBMD, observe the following rules.

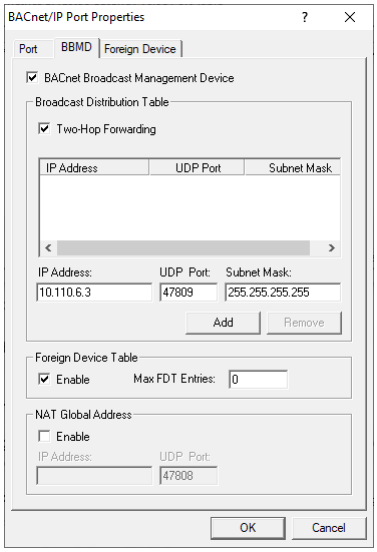
- Configure only one BBMD with the same port number for a single IP subnetwork.
- Assign the same BACnet network number to all BBMDs on the internetwork.
- The BBMD can accept registration from foreign devices or perform BBMD-to-BBMD routing.

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 690.

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.

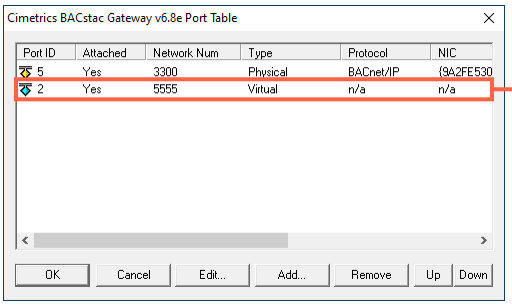
NAT Global Address When using network address translation and port address translation (NAT and PAT), enter the public IP and port number. NAT and PAT are used when the BBMD needs to be accessible from the Internet. The IT department provides the IP address and UDP Port number.

Configure BACstac for Ethernet 8802.3

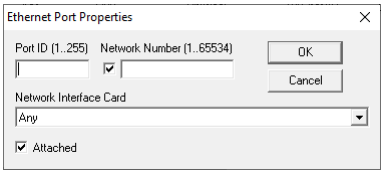
The BACstac driver is automatically installed when KMC Connect is installed. However, the driver must be configured to match the type of BACnet network to which the computer is connected.

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 From **Network Interface Card**, select the physical network adapter that will connect to a LAN.

6 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 this option.

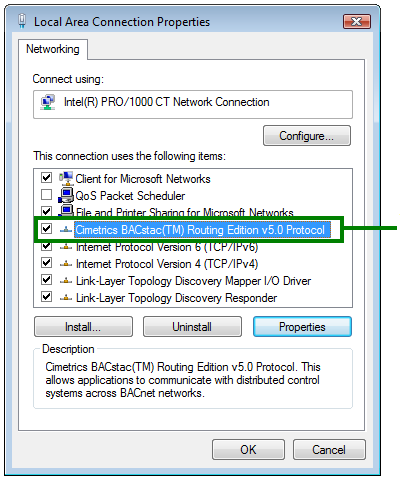
Remove previous versions of the BACstac driver

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 using 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 C: 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 BACnet 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 697
- *Changes to IF THEN* on page 700
- *Deprecated keywords* on page 698
- *File compatibility* on page 700
- *New keywords* on page 701
- *Line numbers and labels* on page 705
- *Local variables* on page 706
- *References to objects in remote devices* on page 706
- *References to properties* on page 706

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 C–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			■
BAC-9300			■
FlexStat BAC-10000 BAC-11000 BAC-120000 BAC-130000 BAC-140000		■	

Deprecated keywords

Keywords in the table *Control Basic deprecated keywords* on page 699 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 697), 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 C–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 705.

- 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 the 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
- ERRORLEVEL
- FLUSH
- HALT
- ISNAN
- LOCALS
- NAN

ALIAS

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 254 for a more detailed description.

BIND

The BIND command 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 261 for a more detailed description.

CONST

Use CONST 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 265 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 279 and *ALIAS* on page 254 for more detailed descriptions.

HALT

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 283 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 289 for a more detailed description.

LOCALS

Use LOCALS 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 292 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 296 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

The COV 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 266 for details and examples.

ENDFUNC

ENDFUNC the end of a user defined Control Basic function.

See the keyword *ENDFUNC* on page 276 and for details and the topic *User-defined functions and procedures* on page 244 for an overview of functions.

ENDPROC

Marks the end of a user defined Control Basic procedure.

See the keyword *ENDPROC* on page 276 for details and the topic *User-defined functions and procedures* on page 244 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 statement executed designates the returned value.

See the keyword *FRETURN* on page 280 and for details and the topic *User-defined functions and procedures* on page 244 for an overview of functions.

FUNCTION

Use FUNCTION to mark 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 280 for an example and the topic *User-defined functions and procedures* on page 244 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 288 for details and an example. For a description of data types, see the topic, *BACnet data types* on page 247.

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 297 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 298 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 311 and *PROCEDURE* on page 312 for details and examples.

PROCEDURE

The PROCEDURE 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 312 for an example and the topic *User-defined functions and procedures* on page 244 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, *BACnet data types* on page 247.

Syntax: *REAL variable[, variable, ...]*

See the keyword *REAL* on page 314 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, *BACnet data types* on page 247.

Syntax: *STRING variable[, variable, ...]*

See the keyword *STRING* on page 327 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 334 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 by 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 (_).
- A label must start with a letter.
- 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 variables with the following statements.

- *INTEGER* on page 288
- *LOCALS* on page 292
- *REAL* on page 314
- *STRING* on page 327

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
```

References to properties

The Next Generation and Generation 5 versions of Control Basic support reading and writing some properties within objects. The common mnemonics are listed in the following table.

See the topic *Mnemonics for BACnet controllers* on page 227 for details on both property and object mnemonics.

Table C–3 Property mnemonics

Property	mnemonic	Action	Comments
High Limit	HL	Read and Write	Intrinsic alarming only
Loop Bias	B	Read and Write	Loop objects only
Loop Derivative	D	Read and Write	Loop objects only
Loop Integral	I	Read and Write	Loop objects only
Loop Proportional	P	Read and Write	Loop objects only
Loop Setpoint	SP	Read and Write	Loop objects only
Low Limit	LL	Read and Write	Intrinsic alarming only
Present Value	PV	Read and Write	Default
Pulse Rate	PR	Read Only	Accumulator objects only
Trend Log Enable	EN	Read and Write	Trend Log objects only
Relinquish Default	RELINQUISH_DEFAULT	Read and Write	Limit writes to changes only

Appendix D: KMC Conquest-compliant controllers

This appendix lists controllers that are compliant and compatible with applications in KMC Connect.

The following KMC Connect applications support controllers listed in this section.

- *Application selection, configuration, and deployment* on page 655
- *Audit application* on page 659
- *VAV Balancing application* on page 663

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

Appendix E: Glossary

A

APDU
Application Layer Protocol Data Unit. An APDU is the significant data in a network packet.

B

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 designated ASHRAE/ANSI 135. BACnet defines how information is exchanged between 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 on the internetwork get the message. Remote Broadcast - All the devices on a remote network get the message. Local Broadcast - All of the devices on the local network receive 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.

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.

C

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.

D

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 numbers range from 0 to 4,194,303 and are assigned to the device during configuration. It is by

reference to the device instance number that data is exchanged between BACnet 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.

H

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.

I

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 BACnet network header (NPCI) if the device is on a remote network.

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, IP or MS/TP.

K

KMDigital
KMDigital is a proprietary line of digital building automation controllers manufactured by KMC Controls, Inc. 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 an EIA-485 network. KMDigital is often abbreviated as KMD.

L

last panel
The highest numbered panel on a KMD 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.

M

main panel
See Tier 1 controller.

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. 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 to 115,200 baud.

N

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 number
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.

O

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.

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.

P

PAD router

A BACnet IP PAD (Packet Assembler and Disassembler) is a special type of router that connects two BACnet IP 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 so that it 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.

point-to-point

In BACnet, point-to-point is a method of data transmission to provide serial communications between two BACnet devices. Typically, it is used for remote and roaming access to BACnet systems, dial-up communications over modems, or a portable computer connection to a controller. PTP is based on the EIA-232 direct point-to-point connection or via dial-up telephone modems.

priority array

BACnet devices use the priority array to control Present Value in certain objects. The array consists of a table with 16 slots arranged in descending priority order. Values written to higher priority slots in the table override those of lower priority. The value in the highest priority slot transfers to the present value property.

Protocol Driver Service

A Protocol Driver Service (PDS) is the link between a building automation system and other services. When data is required from a controller, the PDS retrieves the data, forwards it to the requesting service and, depending upon the type of data, caches it for future use. Each supported protocol requires a corresponding PDS and network connection. The PDS is one of the licensed components of Building Services.

Protocol Implementation Conformance Statement

A Protocol Implementation Conformance Statement (PICS) is a document issued by a manufacturer that specifies exactly which portions of the BACnet Standard a device actually implements.

R

routers, BACnet

BACnet routers connect different types of BACnet networks. The KMC Controls BAC–5051(A)E routers link BACnet 8802-3, BACnet IP, and MS/TP networks.

S

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.

T

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 an OEM equipment interface. 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.

V

variables

Virtual points in the KMD protocal that can represent temperatures, setpoints, offsets, multipliers, or digital values.

W

Who-Is service

The Who-Is service message is sent by a BACnet device to determine the device object identifiers and network addresses for 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.

Index

.
.bac files 428
.bbk files 426
.bnd files 425
.csv files
  archived trend logs 107
  audit log report 91
  Automatic MAC Addressing report 418
  comparing objects 423
  KMD runtime log data 611
  trend log
    BACnet 509
    KMD 615
  trend log multiple 514
.tbvx files 69
.xml files
  archived trend logs 107

A

about
  alarms 133
    BACnet algorithmic 144
    BACnet intrinsic 139
    KMD Control Basic 151
    KMD offline controller 153
  BACnet services 401
  change of value notifications 155
  Code Editor 195
  Control Basic 217
  KMC Connect 33
  Network Manager 41
  objects 401
  OPC 631
  Output Window 53
  properties 401
  Protocol Gateway manager 643
  Protocol Gateway service 643
  Resource Manager 57
  schedules 183
  table view 65
  trend logs 99
ABS 251
abs block 353
ACC 227
access door object 435
accumulator object 458
action
  BACnet PID loop object 484
  KMD PID controller 608
add
  inputs and outputs to expansion panels 420
  objects to device 420
  service to Network Manager list 44
  users to a group
    in Security Manager 83
add block 353
adding 214
address
  IP 421
  KMD Tier 1 574
administrator permissions 81
AI 227
ALARM 252
alarms
  Alarm Monitor bar 167
  archiving 166
  audible 38
  BACnet
    algorithmic 144
    intrinsic 139
  clearing 166
  configuration wizard 136
  configuring BACnet 135
  configuring KMD 153
  disable by network 169
  email server 177
  filtering 168
  filtering by user group 175
  GEST 443
  icons 163
  KMD classes 153
  KMD messages 625
  muting 38
  notification settings 38
  operator notifications 172
  options 38
  overview 133
  printer 177
  summary in BACnet device 166
  viewing 163
  wizards 136
algorithmic alarms
  about 144
  configuring 146
ALIAS 254
analog
  BACnet input objects 461
  BACnet output object 491
  BACnet value objects 518

KMD input point 594
KMD output point 605
KMD variable point 616
AND 257
and block 354
annual schedules
in KMD controllers 589
See BACnet calendar object 441
antiCycle block 354
AO 227
applicable controllers
block programming 339
application
audit 659
commanding VAV airflow 666
configuration 655
custom application wizard 669
deployment 655
selection 655
VAV balancing 663
VAV flow monitoring 666
ARCCOS 257
arccos block 355
arccosine for KMD contollers 266
archiving
alarms 166
trend logs 107
ARCSIN 257
arcsin block 355
arcsine for KMD controllers 323
ARCTAN 258
arctan block 356
arctangent for KMD controllers 328
arithmetic operators 225
array
Control Basic example 591
in KMD controllers 591
length 591
arrays
declaring with
INTEGER 288
LOCALS 292
REAL 314
STRING 327
programming with 246
AS 239
audit
application 659
log 91
log report 91
Auto-Hide 36
automatic
autotune in BACnet PID loops 484
blanking NetSensor for BACnet 529
blanking NetSensor for KMD 604
line numbering 200
MAC addressing 418
automatic log off 38
autotune in BACnet loops 484
AV 227
average
BACnet inputs 463
KMD analog inputs 594
averaging objects 440
AVG 258
avg block 356
AY 239

B

BAC-GET 259
backup
BACnet backup 426
BACstage .bac files 428
entire site 73
files for devices and controllers 62
scheduled device 429
to .bnd file 425
BACnet 684
about services 401
BAC-GET 259
backup procedure 426
BAC-RLQ 260
BAC-SET 261
broadcast management device 690
controllers 567
Ethernet 694
foreign device 690
IP 690
network number 690
objects 435
overview 401
Priority 260
restore 426
BACnet objects 214, 423
BAC-RLQ 260
BAC-SET 261
BACstac driver
configuring 689
removing 695
BACstage
.bac files 428

importing and exporting files 428
balancing application 663
bandwidth
for BACnet networks 684
for KMD networks 685
baud
in devices 443
BBMD configuration 421
BI 227
bias
BACnet PID loop object 484
in BACnet Control Basic 227
in KMD Control Basic 263
KMD PID controller 608
binary
BACnet input objects 466
BACnet output objects 494
BACnet value objects 521
KMD input 595
KMD output 607
KMD variable 617
BIND 261
block programming 339
applicable controllers 339
importing 350
limitations 350
line vs. block 218
local variables 345
macros 347
reference to blocks 353
routines 346
starting the editor 341
subroutines 346
unsupported keywords 350
blocks
abs 353
add 353
and 354
antiCycle 354
arccos 355
arcsin 355
arctan 356
avg 356
call 346
clear 356
close 357
compare 357
compareEqual 358
compareGreater 358
compareGreaterEqual 359
compareLess 359
compareLessEqual 360
compareNotEqual 361
cos 361
counter 362
dec 363
delayOnBreak 363
delayOnMake 362
dewpoint 364
dewpointsi 365
disable 365
divide 366
dom 366-367
enable 368
end 368
enthalpy 368
enthalpysi 369
equalToDiff 370
for 370
Get 344
greaterThanDiff 371
hsel 371
if 372
ifMinus 372
ifPlus 373
inc 373
int 374
integerDivide 374
interval 375
invln 375
lessThanDiff 376
limiter 376
ln 377
lsel 377
macro 347
max 377
min 378
minMaxAvg 378
mod 379
modelnumber 381
month 379
multiply 380
nan 380
netsensorstatus 379
oneShot 381
open 382
or 382
outputoverride 383
paneladdress 383
pi 383
power 383
powerloss 384
program start 384

rem 384
reset 385
rlq 385
rnd 386
scans 386
schedoff 387
schedon 388
select 389
sensoroff 389
sensoron 390
Set 344
sin 391
slope 392
sqr 392
start 392
stop 393
subtract 393
switch 394
tan 394
tbl 395
time 395
timeoff 395
timeon 396
wait 396
xor 397
year 398
bnd backup files 62
BO 227
bookmarks 202
Boolean logic 225
building services
deleting 47
BUILD-NUMBER 262
buttons
NetSensor-BAC controllers 529
NetSensor-KMD controllers 599
BV 227
byte ordering in Modus 596

C

cached values
COV client 156
Calendar object 186
calendar object 441
calibration
BACnet analog input objects 461
BACnet custom device types 461
inputs in KMD controllers 594
NetSensor-BAC controllers 529
NetSensor-KMD controllers 604
STE-9000 NetSensor 535
call block 346
Cbasic table 562
change of value 155
about 155
COV client 155
COV server 155
subscribe configuration pages to object 157
subscribe configuration pages to property 158
subscribe to object 155
subscribe to property 155
subscribe to trend logs 160
subscription list 155
unsubscribing 159
CLEAR 262
clear block 356
CLEAR-COUNT 262
clearing
alarms 166
Control Basic programs 197
Output Window messages 54
clearing want-points
in Tier 1 networks 586
in Tier 2 networks 592
CLOSE 263
close block 357
Code Editor
about 195
adding objects by dragging 214
automatic line numbering 200
changing text font and color 206
live values 210
name/mnemonic button operation 229
options 206
programming 195
select all 201
writing code 196
cold start 416
color
alarm list 163
in Code Editor 206
signal strenght color code 541
combining graphic elements 443
command object 442
commanding airflow 666
communication ports 681
compareEqual block 358
compareGreater block 358
compareGreaterEqual block 359
compareLess block 359
compareLessEqual block 360

compareNotEqual block 361
comparing BACnet objects 423
comparison operators 224
compatibly mode, Niagara 443
compile
  Control Basic programs 197
  options 197
computer
  names 20
  requirements and specifications 18
CON 239
CONBIAS 263
configuration application 655
configuration files, scanning 419
configuring
  alarms
    in BACnet devices 135
    in KMD controllers 153
    operator notifications 172
  BACnet objects 410
  BACnet service 413
  BACstac for Ethernet 694
  BACstac for IP, BBMD and foreign device 690
  building services 46
  Cimetrics BACnet driver 689
  COV notifications
    for configuration pages 157-158
    for trend logs 160
  Design Studio workspace 35
  devices 410
  KMD network 586
  KMD service 584
  notification service 177
  OPC PDS 634
  Protocol Gateway service 651
  sensor port objects 529
  trend logs
    BACnet controller 113
    BACnet device 114
    BACnet PC 117
    KMD controller 121
    KMD device 123
    KMD PC trends 125
  trend service 129
configuring conquest flexstats
  display object 551
connecting
  a Building Service 44
  to an OPC server 633
  to KMDigital
    over the Internet 574
    Tier 1 network 574
    Tier 1 serial 574
    Tier 2 controllers 579
CONPROP 264
conquest flexstat configuration tabs
  display objects 552
CONRATE 264
CONRESET 265
CONST 265
constants in Control Basic 265
contributor permissions 81
Control Basic
  alarms in KMD controllers 151
  automatic line numbering 200
  BACnet tables 562
  editing 196
  expressions 219
  find and replace 202
  for KMD arrays 591
  functions 219
  Generation 5 697
  keywords 251
  KMD runtime logs 611
  KMD weekly schedules 618
  labels 220
  line numbers 220
  line vs. block 218
  live values
    block editor 343
  live values in Code Editor 210
  locking programs 211
  mnemonics
    BACnet 227
    KMD 239
  Next Generation 697
  procedures and functions 244
  scans 218
  statements 219
  undo 201
  using NETPOINT and NETPOINTCACHE 234
  using WAIT statements 233, 237
  variables 226
  version in controllers 697
  want-points 240
  writing offline 198
controllers 709
converting input and output objects 412
copy
  Code Editor 201
  device configuration 410
COS 266

cos block 361
COS-1 266
counter block 362
COV 266
  about 155
  reading and writing off-panel 235
create
  BACnet objects 420
custom
  BACnet device type 461
  KMD units 592
custom application wizard 669
custom table view 69
cut in Code Editor 201

D

DALARM 268
data format in Modbus 596
data type in Modbus 596
data types in OPC tags 637
date
  date and time in KMD controllers 592
  start date in runtime log 611
day
  of month 272
  of week 272
  of year 273
daylight saving time
  in BACnet devices 443
  in KMD controllers 592
DEC 270
dec block 363
declaring
  arrays
    INTEGER 288
    LOCALS 292
    REAL 314
    STRING 327
  constants with CONST 265
  variables
    INTEGER 288
    LOCALS 292
    REAL 314
    STRING 327
DECOM 270
decommissioned 594
default values
  PC trend log interval 48
delayOnBreak block 363
delayOnMake block 362
delete
  building services 47
  inputs and outputs from expansion panels 420
  objects from device 420
deployment application 655
deprecated keywords 698
derivative
  BACnet PID loops 484
  in BACnet Control Basic 227
  in KMDigital Control Basic 264
  KMD PID controller 608
device
  BACnet object 443
  instance 443
  KMD controllers 592
  scheduled backup 429
Device Backup Manager 429
device tables 561
DEWPOINT 270
DEW-POINT 271
dewpoint block 364
DEWPOINTSI 271
dewpointsi block 365
DISABLE 271
disable block 365
discovering devices 43
discovering devices, See Find Devices 408
discovery
  discovering devices 43
  full 43
  limited 43
  options 43
display object 450, 551
divide block 366
docked panes 35
DOM 272
dom block 366
DOW 272
dow block 367
DOY 273
doy block 367
dragging 214
DST
  in BACnet devices 443
  in KMD controllers 592
dynamic access 246

E

editing Control Basic 196
Editing items in RM 58

ELSE 284
email
  alarm notification 172
  backup files 73
  data points with EMAILD 273
  KMD alarm messages
    ALARM 252
    DALARM 268
  messages with EMAILM 274
  runtime logs with EMAILR 275
  server configuration 177
  trend logs with EMAILL 274
EMAILD 273
EMAILL 274
EMAILM 274
EMAILR 275
ENABLE 275
enable block 368
END 276
end block 368
ENDFUNC 276
ENDIF 284
ENDPROC 276
engineering units
  BACnet analog 565
  BACnet binary 565
  KMD analog 621
ENTHALPY 277
enthalpy block 368
ENTHALPYSI 277
ENTHALPY-SI 278
enthalpysi block 369
equalToDiff block 370
ERROR-COUNT 278
ERRORLEVEL 278
errors messages 53
Ethernet
  connection in devices 421
  in BACstac 694
event
  acknowledgment 490
  enrollment object 451
  enrollment summary 160
  processing 490
  summary in BACnet device 166
event log objects 455
example programs 251
  reqlinquish default 231
exception schedules 504
expansion modules 420
exporting
  archived trend logs 107
  audit log 91
  BACstage .bac files 428
  from trend viewer 102
  KMD runtime logs 611
  scheduled backup records 429
  trend log multiple 514
  trend logs
    BACnet 509
    KMD 615
expressions in Control Basic 219

F

file objects 457
file types
  .bac 428
  .bbk 426
  .bnd 425
  .csv
    Automatic MAC Addressing 418
    of audit log 91
    of compared objects 423
  .tbvx 69
filter weight in BACnet inputs 463
filtering
  alarms 168
  alarms by user group 175
  BACnet analog inputs 463
  KMD analog inputs 594
  Output Window messages 54
find
  devices 408
  text in Control Basic 202
firewall 681
firewall exceptions 681
firmware
  file location 93
  upgrades 93
FLASH 337
FLASH over the network 93
floating panes 36
FLUSH 279
for block 370
FOR TO NEXT 279
foreign device configuration 421
FRETURN 280
full discovery 43
FUNCTION 280
function codes in Modbus 596
functions and procedures 244

functions in Control Basic 219

G

Gated Event State Transitions 443
Generation 5 Control Basic 697
GEST 443
get block 344
GOSUB 281
GOTO 282
goto in Code Editor 201
greaterThanDiff block 371
group
  BACnet group object 457
  user groups in Security Manager 83
group roles for security 81
GRP 239

H

halt 197
HALT 283
HANGUP 283
HAVE-TOKEN 283
heartbeat 413
help examples 251
hibernation mode 20
hidden panes 36
hierarchy of operators 223
holiday schedule
  BACnet controllers 441
  KMD controllers 589
HPO-9007 541
HSEL 283
hsel block 371
humidity programing
  NetSensor-BAC controllers 529
  NetSensor-KMD controllers 601

I

icons
  alarms 163
  BACnet devices 408
  KMD controllers 574
  Network Manager 42
  Output Window 53
  STE-9000 NetSensor 535
if block 372
IF- THEN 286
IF THEN (ELSE) 284
IF THEN ELSE 286
IF+ THEN 285
ifMinus block 372
ifPlus block 373
importing
  BACstage .bac files 428
importing block programming 350
IN 239
inactivity timer 38
INC 286
inc block 373
index in array 591
input objects
  accumulator 458
  analog 461
  BACnet pulse 458
  binary 466
  changing type 412
  creating and deleting 420
  multistate 469
input tables 561
installing
  computers 20
  KMC Connect 20
INT 287
int block 374
INTEGER 288
integerDivide block 374
integral
  BACnet PID loops 484
  in BACnet Control Basic 227
  in KMDigital Control Basic 265
  KMD PID controller 608
Intellisense filter 200
interprotocol mapping 596
INTERVAL 289
interval block 375
INVLN 289
invln block 375
IP address 421
ISNAN 289
ISSTRING 290

K

keyboard shortcuts
  Code Editor 201
keywords 251
  deprecated 698
  Intellisense filter 206
  prompting 200
KMC Connect 33
KMD
  backup files 62

system name 574
KMD-5559 for KMD Tier 2 579
KMD-5576 for KMD Tier 2 579

L

labels 220
leap year 294
length 591
array 591
runtime log 611
lessThanDiff block 376
LET 290
license
for KMD service 584
life safety objects
point 472
zone 476
limitations of block programming 350
limited discovery 43
limiter block 376
limits
KMD want-points 241
off-panel points 232
line numbers 220
live values
block editor 343
Code Editor 210
LN 291
ln block 377
LN-1 291
load control objects 480
local variables 226, 345
LOCALS 292
locking programs
how to 211
properties 443
lockout application 38
logarithm
invln (inverse natural) block 375
ln (natural) block 377
logging
KMD 121
Ouput Window 54
logs, runtime 611
LOOP 227
loop
BACnet PID loop object 484
PID loops in KMD controllers 608
LSEL 293
lsel block 377

M

MAC address
in BACnet 418
in devices 443
macro blocks
antiCycle 354
delayOnBreak 363
delayOnMake 362
equalToDiff 370
greaterThanDiff 371
limiter 376
minMaxAvg 378
reset 385
reusing 347
saving 347
select 389
slope 392
switch 394
user defined 347
MAMA report 418
manual change
analog value 518
binary value 521, 525
output 494
weekly schedule 618
MAX 293
max block 377
max master 443
messages in Output Window 53
MIN 294
min block 378
minMaxAvg block 378
mnemonics
BACnet object 227
BACnet properties 227
KMD 239
MOD 294
mod block 379
Modbus
data format 596
data type 596
function codes 596
Multiplier 596
offset 596
register 596
MODBUSTRANSFER 295
MODELNUMBER 295
modelnumber block 381
modulus
block 379

keyword 294
MONTH 296
month block 379
motion sensing
NetSensor-BAC controllers 529
NetSensor-KMD controllers 603
MSI 227
MSO 227
MSV 227
Multiplier in Modbus 596
multiply block 380
multistate objects
input 469
output 498
value 525
mute alarms 38

N

names
alarm recipient operators 172
names for computers 20
NAN 296
nan block 380
Near Field Communication (NFC)
for service and network 52
for single devices 443
NEG_INFINITY 296
NETPOINT 297
NETPOINTCACHE 298
NetSensor
Conquest 535
STE-9000 535
NetSensor-BAC controllers
auto-blanking 529
buttons 529
calibration 529
humidity 529
motion sensing 529
programming 529
setpoint 529
temperature 529
time 529
verifying status 529
NetSensor-KMD controllers
auto-blanking 604
buttons 599
calibration 604
humidity 601
motion sensing 603
programming 600
setpoint 601
temperature 601
time 602
verifying status 602
NETSENSORSTATUS 300
NETSENSOR-STATUS 300
netsensorstatus block 379
network
FLASHing controllers 93
ports 681
network address translation 574
Network Manager 41
display settings 48
icons 42
opening 42
options 48
scanning for uninitialized files 419
network number 690
network status
KMD Tier 1 586
KMD Tier 2 592
networks 49
new
security group in Security Manager 83
Next Generation Control Basic 697
NFC
for service and network 52
for single devices 443
Niagara compatibility mode 443
NOT 301
not block 381
notification class object 490
notifications
BACnet algorithmic 144
BACnet change of value 155
BACnet intrinsic 139
KMD controllers 151
NPAGE
in Tier 1 controllers 301
message strings in alarms 252, 268

O

object locate 39
objects
about 401
access door 435
accumulator 458
analog input 461
analog output 491
analog value 518
averaging 440
binary input 466

binary output 494
binary value 521
calendar 186, 441
command 442
creating and deleting 420
device 443
event enrollment 451
event log 455
file 457
group 457
life safety point 472
life safety zone 476
load control 480
loop 484
multistate input 469
multistate output 498
multistate value 525
notification class 490
program 501
schedule 504
sensor port 529
structured view 508
trend 509
trend log multiple 514
offline programming 30
off-panel points
BACnet limits 232
reading 232
using COV 235
using NETPOINT and NETPOINTCACHE 234
with INTERVAL 237
with WAIT 233, 237
writing 232
Offset in Modbus 596
ON GOSUB 302
ON GOTO 303
ON-ALARM 304
ONERROR 305
ON-ERROR 305
oneShot block 381
OPC 631
connecting to servers 633
objects 637
quality 637
tags 637
timestamp 637
OPEN 306
open block 382
opening
Control Basic programs 196
Network Manager 42
Output Window 54
Resource folder 63
Resource Manager 58
operators
arithmetic 225
Boolean 225
comparison 224
precedence 223
relational 224
options
alarm popup 38
audible alarm 38
Code Editor 206
discovery 43
for the workspace 38
Network Manager 48
Output Window 54
OR 307
or block 382
OUT 239
output
KMD PID control loop 608
weekly schedule 618
output objects
analog 491
binary 494
changing type 412
creating and deleting 420
multistate 498
Output Window 53
icons 53
OUTPUTOVERRIDE 307
OUTPUT-OVERRIDE 308
outputoverride block 383
override of weekly schedule 618

P

PANELADDRESS 308
PANEL-ADDRESS 308
paneladdress block 383
panes
docked 35
floating 36
hidden 36
password 77
in KMD controllers 592
STE-9000 NetSenSensor 535
to reinitialize BACnet device 416
paste in Code Editor 201
peer-to-peer transfer
BACnet 232

KMD 240
permissions
changing for a group 83
contributor 81
reader 81
web administrator 81
PHONE 309
PI 310
pi block 383
PID loops
BACnet PID loop object 484
KMD controllers 608
pinned 36
plus mode in KMD controllers 592
point-passing
BACnet 232
KMD 240
points
KMD analog input 594
KMD analog output 605
KMD analog variable 616
KMD annual schedule 589
KMD binary input 595
KMD binary output 607
KMD binary variable 617
KMD interprotocol 596
KMD PID controllers 608
KMD PID loops 608
KMD runtime log 611
KMD system groups 613
KMD tables 614
KMD trend logs 615
ports
communications 681
for BACnet 684
for KMD networks 685
in firewalls 681
POS-INFINITY 310
power block 383
power settings 20
POWERLOSS 311
POWER-LOSS 311
powerloss block 384
precedence of operators 223
PRETURN 311
PRG 239
printer, alarm 177
priority array 402
priority for writing
Control Basic syntax 231
macro block 347
set block 344
syntax 231
priority levels
BACnet standard levels 402
Control Basic default 402
PROCEDURE 312
procedures and functions 244
process identifier 490
program start block 384
programming examples
array in KMD controller 591
NetSensor-BAC controllers 529
NetSensor-KMD controllers 600
PID loop in KMD controller 608
runtime logs in KMD controllers 611
weekly schedules in KMD controllers 618
programming notation 222
programs 611
properties
about 401
proportional
BACnet PID loops 484
in BACnet Control Basic 227
in KMD Control Basic 264
KMD PID controller 608
Protocol Gateway
about 643
configuring 651
create a process 647
delete a process 647
opening Protocol Gateway Manager 645
restart processes 647
test a process 647
public IP address 574
pulse input 458

R

radians 310
random numbers 316
rate in KMD PID control loops 608
RD 227
READ-CONBIAS 312
READ-CONPROP 313
READ-CONRATE 313
READ-CONRESET 313
reader permissions 81
reading
system groups from a device 592
system groups from device, network or service 613
reading properties from other devices 232

REAL 314
redo in Code Editor 201
regenerate networks 415
regular expressions 202
reinitialize device 416
relational operators 224
relinquish
  default 231
  priority 231
  priority level keyword 315
REM 315
rem block 384
remote access 683
renaming objects NM 49
renaming objects RM 64
renumber Control Basic lines 201
repeats in PID loops 608
replace
  text in Control Basic 202
reports
  audit log 91
  MS/TP Automatic MAC Addressing 418
requirements
  computers 18
reset block 385
reset in PID control loops 608
Resource folder 63
Resource Manager 57
restart
  BACnet device 416
  BACstac driver
    from Network Manager 51
restoring
  BACnet restore 426
  controller configuration 59
  site from backup file 74
RETURN 315
reusing macro blocks 347
RLQ 315
rlq block 385
RND 316
rnd block 386
routine start block 386
routines 346
RT 239
run
  Control Basic programs 197
  set to Control Basic to run in KMD controllers 611
runtime logs
  programming 611
  start date 611
  starts today 611
  total starts 611

S

saving
  trend logs
    from trend viewer 102
saving macro blocks 347
scanning for uninitialized files 419
scans 218
SCANS 316
scans block 386
SCHED 227
SCHEDOFF 317
SCHED-OFF 318
schedoff block 387
SCHEDON 318
SCHED-ON 319
schedon block 388
schedules
  about 183
  annual in KMD controllers 589
  annual, See calendar object 441
  BACnet schedule object 504
  device backup 429
  exception 504
  holiday 441
  KMD
    weekly schedule point 618
  object 504
  override 504
  SCHED-OFF 318
  SCHEDON 318
  weekly point 618
security 77
  contributor 81
  group roles 81
  groups 83
  reader 81
  web admin administrator 81
Security Manager 77
select all 201
select block 389
selection application 655
sensor objects
  NetSensor
    BACnet 529
    Conquest 535
  wireless gateway 541
SENSOROFF 319

SENSOR-OFF 320
sensoroff block 389
SENSORON 321
SENSOR-ON 322
sensoron block 390
served web pages 50
services
  adding to Network Manager 44
  configuring 46
set block 344
setpoint
  in KMD PID control loop 608
  NetSensor-BAC controllers 529
  NetSensor-KMD controllers 601
shedding load with load control object 480
shortcut keys
  Code Editor 201
Show By in Output Window 54
side panes 39
SIN 324
sin block 391
SIN-1 323
site
  add new 27
  back up 73
  close 30
  management tools 33
  offline programming 30
  open existing 29
  restoring from backup file 74
sleep mode 20
slope block 392
Sort By in Output Window 54
sorting
  Output Window messages 54
  Output Window options 54
specifications
  computers 18
SQR 324
sqr block 392
START 325
start block 392
start date in runtime log 611
starts
  today in runtime logs 611
  total in runtime log 611
startup 24-25
statements in Control Basic 219
STATUS 326
STEP 279
STOP 326
stop block 393
STRING 327
structured view objects 508
subscribe to object 155
subscribe to property 155
subscription list 155
subtract block 393
summary,event enrollment 160
switch block 394
system groups 613
  reading from device 592
system name for KMD networks 574

T

table view
  about 65
  custom views 69
  group of objects 66
tables 561
  BACnet Control Basic 562
  BACnet device 561
  BACnet input 561
  Cbasic 562
  KMD controllers 614
  TBL in Control Basic 329
tags in OPC servers 637
TAN 328
tan block 394
TAN-1 328
TBL 329
tbl block 395
temperature
  NetSensor-BAC controllers 529
  NetSensor-KMD controllers 601
testing
  gateway processes 647
text
  font and color in Code Editor 206
time
  BACnet
    DST in devices 443
    setting in devices 417
    synchronization 417
    synchronization interval 413
    UTC in systems 417
  KMD networks 584, 587
  NetSensor-BAC controllers 529
  NetSensor-KMD controllers 602
  OPC timestamp 637
  want-point transfer 241

TIME 330
time block 395
TIMEOFF 332
TIME-OFF 332
timeoff block 395
TIMEON 333
TIME-ON 334
timeon block 396
timeout 38
TL 227, 239
toolbar
  Network Manager 42
tools
  site management 33
TOSTRING 334
total on time, runtime log 611
TPAGE
  in Tier 1 controllers 335
  message strings in alarms 252, 268
trend logs
  archiving 107
  change of value 117
  configuration wizard 110
  configuring
    BACnet controller 113
    BACnet device trend logs 114
    BACnet PC trend logs 117
    KMD controller 121
    KMD device trend logs 123
    KMD PC trend logs 125
  creating with trend log multiple object 514
  creating with trend object 509
  in KMD controllers 615
  trend log multiple object 514
  viewing
    BACnet controller trends 509
    KMD controller trends 615
  viewing controller 106
  wizard 110
trend object 509
types 247

U

UNACK 335
undo 201
uninstall 21
units
  standard analog units 565
  standard binary units 565
unsubscribing from COV subscriptions 159
upgrading firmware 93
user groups in Security Manager 83
user interface 551
user name 77
USER-A 335
UTC (Universal Coordinated Time) 417, 443

V

value in array 591
value objects
  analog 518
  binary 521
  multistate 525
VAR 239
variable point
  KMD analog 616
  KMD binary point 617
variables 226
  declaring with
    INTEGER 288
    LOCALS 292
    REAL 314
    STRING 327
variables, local 345
VAV
  balancing application 663
  commanding airflow 666
  flow monitoring 666
viewing
  alarms 163
  COV subscriptions 160
  event enrollment summary 160
  from trend viewer 102
  groups of objects 65
  notification summary in BACnet 166
  trend logs
    controller logs 106

W

WAIT 336
wait block 396
want-points
  clearing 592
  initial value 241
  KMD mnemonics 239
  limitations 241
  requesting 240
  total 241
  transfer time 241
  using COV 235
  using INTERVAL 237
  using NETPOINT and NETPOINTCACHE 234

using WAIT 233, 237
warm start 416
web administrator 81
weekly schedules 618
in BACnet schedule objects 504
KMD weekly schedules 618
programming example 618
wireless
color code 541
Control Basic 547
gateway 541
sensors 541
within block 397
wizards
alarm configuration 136
analog input object 673
analog output object 677
Bacnet input objects 673
Bacnet output objects 673
binary input object 676
custom application 669
trend configuration 110
word ordering in Modbus 596
workspace 39
workspace in Design Studio 35
WRITE-TO-FLASH 337
writing
programs 219
properties to other devices 232
WS 239
WS-OFF 337
WS-ON 338

X

XOR 338
xor block 397

Y

YEAR 338
year block 398

Z

zero-length files 419
zoom in and out
trend logs 106