Honeywell

Experion PKS

Bristol Babcock Interface Reference

EPDOC-XXX8-en-431A
February 2015

Release 431

Document	Release	Issue	Date
EPDOC-XXX8-en-431A	431	0	February 2015

Disclaimer

This document contains Honeywell proprietary information. Information contained herein is to be used solely for the purpose submitted, and no part of this document or its contents shall be reproduced, published, or disclosed to a third party without the express permission of Honeywell International Sàrl.

While this information is presented in good faith and believed to be accurate, Honeywell disclaims the implied warranties of merchantability and fitness for a purpose and makes no express warranties except as may be stated in its written agreement with and for its customer.

In no event is Honeywell liable to anyone for any direct, special, or consequential damages. The information and specifications in this document are subject to change without notice.

Contents

Planning considerations for installing and configuring Bristol Babcock controllers

This reference describes how to set up, configure, and test Bristol Babcock controller communications with the server. Before you can use Quick Builder to define Bristol Babcock channels and controllers, you will need to use the special utility **bbhdw**.

Revision history

Revision	Date	Description
A	February 2015	Initial release of document.

How to use this guide

The following steps show the order in which the controller interface should be configured. Complete each step before starting the next.

Steps for connecting and configuring a Bristol Babcock controller.

Steps	Go to
Set up Bristol Babcock controller	Architectures for Bristol Babcock
Install ACCOL tools on server	Installing ACCOL tools on the server
Install initial load onto Bristol Babcock nodes via the Portable Engineers Interface (PEI)	Installing the initial load on the Bristol Babcock node
Connect all cables and modems	Communication settings for Bristol Babcock
Create a *`netfile.dat`* file using ACCOL utility, **nettop5**	Defining the network
Run the **bbhdw** utility to produce a hardware definition file	Defining the connections to controllers
Using the **bbhdw** output, create a channel in Quick Builder	• Bristol Babcock channel and controller reference • *Quick Builder User's Guide*
Using the **bbhdw** output, create a controller in Quick Builder	• Bristol Babcock channel and controller reference • *Quick Builder User's Guide*
Download channel and controller definitions to the server	*Quick Builder User's Guide*
Test communications with the server	Testing Bristol Babcock communications with the server
Create and download ACCOL load files using **aic5** and **toolkit5**	Using ACCOL load files to define scanning and control strategies
Define points in Quick Builder, and then download them to the server	Defining a Bristol Babcock address for a point parameter

Related topics

"Devices supported by the Bristol Babcock interface" on page 7

"Other documentation for Bristol Babcock" on page 8

“Architectures for Bristol Babcock” on page 9

“Installing the initial load on the Bristol Babcock node” on page 10

“Communication settings for Bristol Babcock” on page 11

“Defining the network” on page 13

“Architectures for Bristol Babcock” on page 9

“Installing ACCOL tools on the server” on page 9

“Defining the connections to controllers” on page 16

“Bristol Babcock channel and controller reference” on page 15

“Testing Bristol Babcock communications with the server” on page 39

“Using ACCOL load files to define scanning and control strategies” on page 25

“Defining a Bristol Babcock address for a point parameter” on page 30

Devices supported by the Bristol Babcock interface

Generally, the server supports any device that communicates via the Bristol Babcock Synchronous Asynchronous communication protocol (BSAP) and fits in the Network 3000 architecture.

Only the DPC 3330 has been qualified with the Experion server.

Other documentation for Bristol Babcock

The following Bristol Babcock document describes the address space of all the devices supported by this interface in great detail. It also provides information about the physical configuration of the controllers. Reading this document is also useful when installing the interface.

- *DPC 3330 Instruction Manual*

The following Bristol Babcock documents might also be useful when installing and configuring the DPC 3330:

- *Network 3000 Communication User's Guide*
- *RDC 3350 User's Manual*
- *Bristol Babcock ACCOL Reference* and ACCOL Tools Binders
- *ACCOL II Interactive Compiler (AIC) Manual*
- *ACCOL II Reference Manual*

From Honeywell:

- *Software Installation User's Guide*
- *Hardware and Point Build Reference*

Architectures for Bristol Babcock

The following architectures are possible for Bristol Babcock:

- Direct serial connection via a Bristol Babcock Data Concentrator using RS-232 and BSAP protocol
- Multi-dropped on a single serial link using RS-485

The RS-232 connection is between a serial port on the server and the serial interface on the Bristol Babcock device. The BSAP protocol is fully described in the *Network 3000 Communications User's Guide.*

The RS-485 connection requires a RS-232 to RS-485 Black Box converter or Stallion EasyConnection adapter between the server serial port and the RS-485 network. See the *Software Installation User's Guide* for information about using these devices with the server.

Related topics

"Planning considerations for installing and configuring Bristol Babcock controllers" on page 5

Installing ACCOL tools on the server

Some ACCOL tools are required to run on the server platform. These are:

- **aic5** (produces ACO files)
- **aclink5** (produces ACL files)
- **nettop5** (produces `netfile.dat` file)
- **toolkit5** (downloads ACL files)

Install these tools in the `\accol` folder. ACCOL tools can be started from the Windows Explorer or a command prompt from any working folder, for example: `\accol\aclink5`, and `\accol\nettop5`.

Related topics

"Planning considerations for installing and configuring Bristol Babcock controllers" on page 5

Installing the initial load on the Bristol Babcock node

A bootstrap process is required to configure the communications parameters. This is achieved by using a PEI to download a small ACCOL load program, which sets up the communication ports. This is done before you connect the Bristol Babcock devices to the server. See the *ACCOL II Interactive Compiler and Toolkit* manuals for instructions on ACCOL programming and PEI operation.

To create and download an initial load on the PEI

1. Using **aic5** software on the PEI to create the initial load, set the communication port to the correct baud, parity, and protocol for server communication, particularly:
   - Baud (9600, 19,200, or 38,400)
   - Parity (None)
   - Protocol (RS-232)
2. Use **aclink5** to link the ACO file and create an ACL file.
3. Use **toolkit5** to download the ACL file to the Bristol Babcock node.

   Write down the communication characteristics of local nodes (that is, nodes on network level 1 that communicate directly with the server) in preparation for defining channel connections to controllers in Quick Builder.

Communication settings for Bristol Babcock

After downloading the initial load to all Bristol Babcock nodes, connect the cables. The server to Bristol Babcock connection usually uses an RS-232 cable. “Figure 1: RS-232 connect using D15 to D25” shows the pinouts for an RS-232 connection to a server-end D25 serial port. Bristol Babcock devices use a 15-pin D connector.

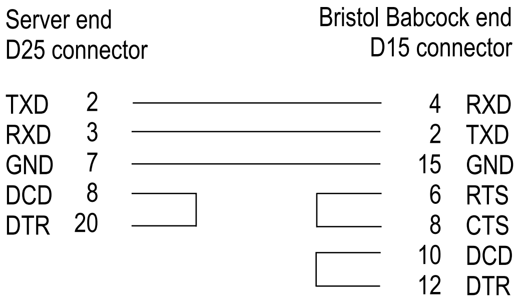

Figure 1: RS-232 connect using D15 to D25

“Figure 2: RS-232 connect using D15 to D9” shows the pinouts for an RS-232 connection to a server-end D9 serial port.

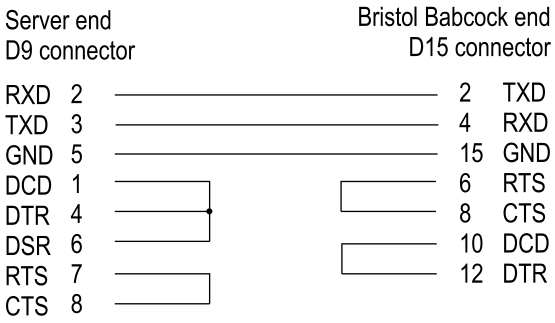

Figure 2: RS-232 connect using D15 to D9

The server can connect to an RS-485 network using additional hardware. At the server end an RS-232/RS-485 converter or adapter is required. At the controller end, Network Interface Boxes (NIB), supplied by Bristol Babcock, are required. The *DPC3330 Instruction Manual* (CI3330) gives the configuration for the NIB hardware.

“Figure 3: RS-485 network with NIBs and an RS-485 converter” shows the pinouts for RS-485 network with NIBs and a converter.

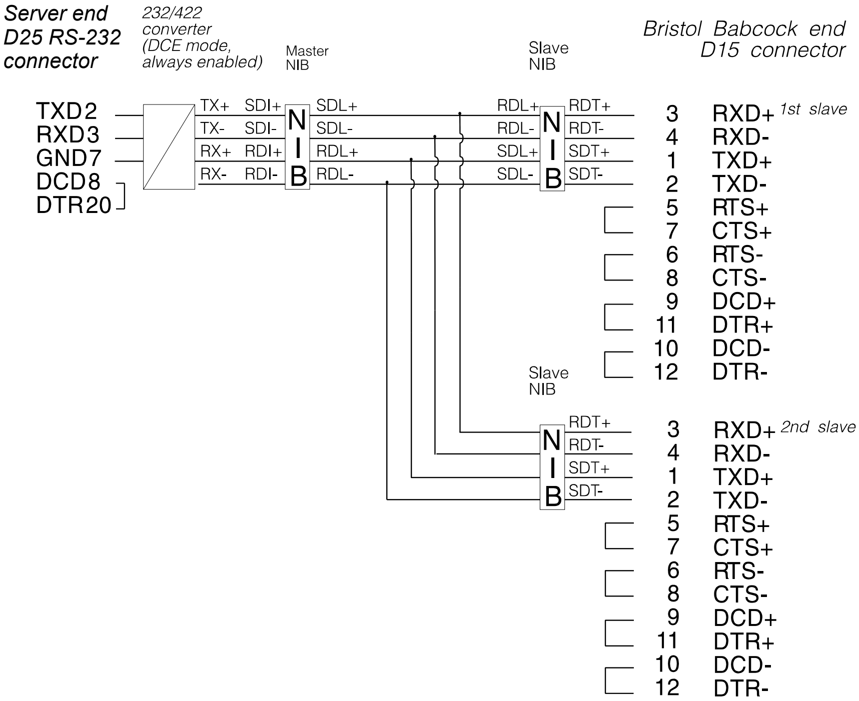

Figure 3: RS-485 network with NIBs and an RS-485 converter

Defining the network

Define the network topology using the Bristol Babcock utility **nettop5**. See the appropriate Bristol Babcock documentation for details on how to use this program.

The output of **nettop5** is a file called *netfile.dat*, which is used in two ways:

- Firstly, it is downloaded to local nodes by the server so that those nodes can send global messages.
- Secondly, it is used as an input for the **bbhdw** utility and the resulting file is used for defining connections to controllers in Quick Builder.

Related topics

"Downloading netfile.dat to the network" on page 13

"Defining the connections to controllers" on page 16

Downloading netfile.dat to the network

By default, the *netfile.dat* file is saved to *\accol*. It is then automatically downloaded from this folder by the server if the system has a single network (channel).

If there are multiple networks (and therefore channels) connected to the server, for the correct *netfile.dat* file to be downloaded automatically it must be in:

```
server\data\bb\chnccc\
```

where

ccc is the channel number of the Bristol Babcock Network. For example, `chn001`.

Copy the *netfile.dat* file produced in *\accol* to a file of the same name in *server\data\bb\chnyyy\* for each channel. The server will look in these folders first.

Related topics

"Defining the network" on page 13

Bristol Babcock channel and controller reference

This section describes the configuration and addressing information specific to Bristol channels and controllers.

In addition to the information contained in this reference, and for help to build channels and controllers, see the section titled "Building controllers or channels" in the *Quick Builder User's Guide*.

Related topics

“Defining the connections to controllers” on page 16
“Main properties for a Bristol Babcock channel” on page 18
“Port properties for a Bristol Babcock channel” on page 20
“Main properties for a Bristol Babcock controller” on page 22
“Using ACCOL load files to define scanning and control strategies” on page 25
“Optimizing Bristol Babcock scanning performance” on page 27
“Planning considerations for installing and configuring Bristol Babcock controllers” on page 5

Defining the connections to controllers

The process involved in defining connections to controllers is shown.

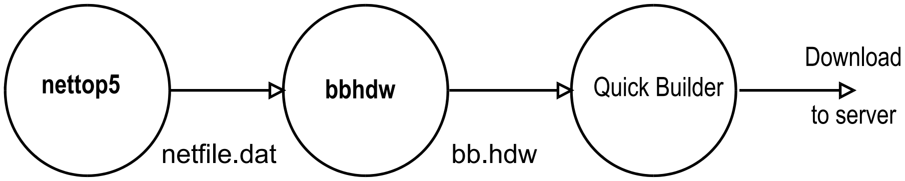

Figure 4: Defining connections to controllers

The essential steps for configuring Bristol Babcock hardware and defining the communication channels are:

1. Convert the Bristol Babcock `netfile.dat` file (containing a complete definition of the network) to a hardware definition file. The utility used is **bbhdw**. The first step in the total process, creating the `netfile.dat` file, is described in the topic titled "Defining the network."
2. Use the values obtained from the **bbhdw** output file to define Bristol Babcock channels and controllers in Quick Builder.
3. After the Bristol Babcock channel and controllers are defined in Quick Builder and the definitions downloaded to the server, test the communications between the server and the controller using the **bbtst** diagnostic utility. See the topic titled "Testing Bristol Babcock communications with the server" for more information.

Related topics

“Planning considerations for installing and configuring Bristol Babcock controllers” on page 5

“Defining the network” on page 13

“Testing Bristol Babcock communications with the server” on page 39

Using the BBHDW utility

This utility converts the `netfile.dat` file to a hardware definition file suitable for hardware building. The `netfile.dat` file is produced by the Bristol Babcock **nettop5** program and is a complete definition of the network.

The syntax for **bbhdw** is:

```
bbhdw [options]
```

Option	Description
-i nrtfile	Input the NRT source file. *netfile.dat* is the default— the *.dat* extension does not need to be specified.
-o hdwfile	Output hardware source file. *netfile.hdw* is the default. The *.hdw* extension does not need to be specified.
-z n	Zone/area number. Take Bristol Babcock nodes assigned to zone/area *n* only. Defaults to all nodes.
-c cc	Channel number. Assigns all controllers to channel number *cc*. The default is *1*.
-r rrr	Starting controller number is *rrr*. The default is *1*.

Examples

Some example calls:

```
bbhdw
```

Extract all nodes from *netfile.dat* and create *netfile.hdw* using channel 1 and two controllers per node, (one for signals and one for arrays) starting at controller 1.

```
bbhdw -o bbchn
```

As above, except *bbchn.hdw* is created.

```
bbhdw -z 3
```

Extract all nodes in area/zone *3* only.

```
bbhdw -c 10 -r 5
```

Put all nodes on channel *10*, beginning with controller *5*.

Output file example

The **bbhdw** output file consists of the following entries:

```
DEL CHNcc
ADD CHNcc SERIAL PORT=COM1 BAUD=9600 PARITY=none READ=10
DEF CHNcc BB NAME=BBabcock MARG=50 FAIL=100 NRTSHIFT=s NRTMASK=m
   NRTVER=v
```

Then for each node, a controller of each type (signals and arrays) is added (*sss = rrr+1*):

```
DEL RTUrrr
DEL RTUsss
DEF RTUcc.rrr 0 NODENAME=dpc1 NAME=dpc1_sig ID=addr/level
    NWID=gladdr MARG=25
FAIL=50
DEF RTUcc.sss 1 NODENAME=dpc1 NAME=dpc1_arr ID=addr/level
    NWID=gladdr MARG=25 FAIL=50
```

Main properties for a Bristol Babcock channel

Use the Main tab to enter the basic channel properties for a Bristol Babcock channel.

If the *DEF CHNcc* line produced was:

```
DEF CHN01 BB NAME=BBabcock MARG=50 FAIL=100 NRTSHIFT=10
    NRTMASK=1f NRTVER=225
```

you would enter:

- `10` for NRT Shift
- `1f` for NRT Mask, and
- `225` for NRT Version

For information about how to create a channel, see the topic titled "Building controllers and channels" in the *Quick Builder User's Guide*.

Property	Description
Name	The unique name of the channel. A maximum of *10* alphanumeric characters (no spaces or double quotes). Note: In Station displays, underscore characters (_) appear as spaces.
Description	(Optional) A description of the channel. A maximum of *132* alphanumeric characters, including spaces.
Associated Asset	The Tag Name of the Asset to be associated with the alarm group.
Marginal Alarm Limit	The communications alarm marginal limit at which the channel is declared to be marginal. When this limit is reached, a high priority alarm is generated. To change the priority of the alarm system wide, see the topic titled "Configuring system alarm priorities" in the *Server and Client Configuration Guide*. To change the priority of the alarm for one channel, see the topic titled "About configuring custom system alarm priorities for an individual channel or controller" in the *Server and Client Configuration Guide*. A channel barometer monitors the total number of requests and the number of times the controller did not respond or response was incorrect. The barometer increments by two or more, depending on the error, and decrements for each good call. To calculate an acceptable marginal alarm limit, use the formula: Square root of the number of controllers on the channel × Marginal Alarm Limit defined on those controllers (Normally, you specify the same value for all controllers on a channel). For example, if there are 9 controllers on the channel and their Marginal Alarm Limit is set to *25*, the value would be *3* (which is the square root of 9) *x 25 = 75*.
Fail Alarm Limit	The communications alarm fail limit at which the channel is declared to have failed. When this barometer limit is reached, an urgent alarm is generated. To change the priority of the alarm system wide, see the topic titled "Configuring system alarm priorities" in the *Server and Client Configuration Guide*. To change the priority of the alarm for one channel, see the topic titled "About configuring custom system alarm priorities for an individual channel or controller" in the *Server and Client Configuration Guide*. Set this to double the value specified for the channel Marginal Alarm Limit.
NRT Shift	*NRTSHIFT=* Node Routing Table Level 0 shift count. This is a network parameter. For more information see the *Bristol Babcock Network 3000 User's Guide*. This parameter is produced by the **bbhdw** utility. This parameter is not normally changed.
NRT Mask	*NRTMASK=* Node Routing Table Level 0 bit mask. This is a network parameter entered as a hexadecimal number. For more information see the *Bristol Babcock Network 3000 User's Guide*. This parameter is produced by the **bbhdw** utility. This parameter is not normally changed.

Property	Description
NRT Version	`NRTVER=` Node Routing Table version. This is a network parameter. This number will be produced by the utility **bbhdw**. This parameter is not normally changed.
Exception Poll Period	`POLLPER=` The default value is 10 seconds. All controllers will be put on an exception poll for the scan period closest to this value. This parameter is not produced by the **bbhdw** utility. See the topic titled "Optimizing Bristol Babcock scanning performance" for more information.
Control Timeout	`RESPONSE=` Response time for a device on the network. This value is multiplied by the network level to get an absolute response time between the server and the remote device. The default is *10* seconds. Should be set to the worst case RDB round trip time for any level of the network.
Connect Timeout	The length of time that the server attempts to connect to the controller. The server will stop trying to connect to the controller once the timeout period passes. The default value *10* seconds. Use the default value unless the communications line has a high error rate, or unless you are using modems.
Read Timeout	The length of time that the server will wait for a reply from the controller. The server will stop waiting once the timeout period passes. The default value is *2* seconds. Use the default value unless the communications line has a high error rate, or unless you are using modems.
Write Delay	If the channel is on a serial port, the length of time (in milliseconds) that the server waits before writing to any controller on the channel. The default value is *10* milliseconds. A write delay is usually specified only if: • The server communicates to the controller over a half-duplex radio link and the radio system requires time to key in each direction before the server or controller can send data. • The radio system implements *RTS/CTS* handshaking. If there is a communications problem and the controller does not respond to writes from the server, try changing this setting to *11* milliseconds or more. This should allow the controller enough time to become ready to receive data from the server.
Item Type	The type of channel specified when this item was created.
Last Modified	The date and time the channel properties were modified.
Last Downloaded	The date and time the channel was last downloaded to the server.
Item Number	The unique item number currently assigned to this channel, in the format `CHNcc`, where `cc` is the channel number. You can change the item number if you need to match your current server database configuration. The number must be between *01* and the maximum number of channels allowed for your system. For more information about setting the maximum value, see the topic titled "Adjusting sizing of non-licensed items" in the *Supplementary Installation Tasks Guide*.

Related topics

"Optimizing Bristol Babcock scanning performance" on page 27

Port properties for a Bristol Babcock channel

Select the Port Property sheet and enter the channel communication settings. Use the output of **bbhdw** (the *ADD CHNcc* line) previously generated.

The Port tab defines the communication-related properties for a channel. The Port Type for Bristol Babcock can be:

- *Serial*. A serial communications interface, such as RS-232. See the section below titled "Serial Port properties."
- *TerminalServer*. A communications link that enables controllers with a serial interface to be connected to a LAN. See the section below titled "Terminal Server Port properties."
- *LANVendor*. Not applicable for Bristol Babcock.

Serial port properties

The Serial Port settings must match the output of the **bbhdw** file. For example, if the output of **bbhdw** generates an *ADD CHNcc* line as:

```
ADD CHN01 SERIAL PORT=COM1 BAUD=9600 PARITY=NONE READ=10
```

then enter:

- *Serial* as the Port Type
- *COM1* as the Serial Port Name
- *9600* as the Baud
- *None* as the Parity

Property	Description
Serial Port Name	The device name of the serial port.
Baud	The number of data bits per second. The default is *9600*.
Number of Data Bits	The number of data bits used for transmission. The default is *8*.
Stop Bits	The number of stop bits used for transmission The default is *1*.
Parity	Defines parity verification of each character and must match configuration on the end device. The default is *NONE*.
Checksum	The type of checksum error detection used for the port. Not applicable for this channel. Select *NONE*.
XON/XOFF	The type of XON/XOFF software flow control used to stop a receiver from being overrun with messages from a sender. The types are: • *Input* (use XON/XOFF to control the flow of data on the receive line) • *None* (default) • *Output* (use XON/XOFF to control the flow of data on the transmit line)

Property	Description
RS-232	These options are applicable to the RS-232 link: • **Enable RTS/CTS flow control**. Select this if you want to use RTS/CTS for flow control to stop a receiver from being overrun with messages from a sender. • **Detect DCD**. Select this if the Data Carrier Detect communication status line of the COM port requires monitoring (usually when using modem or microwave linking). When selected, the communications fails if the desired COM status line is not high—for example, on a dial-up link connection for a modem. • **Detect DSR**. Select this if the Data Set Ready communication status line of the COM port requires monitoring (usually when using modem or microwave linking). When selected, the communications fails if the desired COM status is not achieved.
RS-485	Not applicable to Bristol Babcock connections, even if RS-232/RS-422 converters are used.
Protocol	The protocol to be used to communicate on this Bristol Babcock channel. All controllers on the channel must be configured for this protocol. The default is *BSAP*.

Terminal Server port properties

If a terminal server is to be used to communicate with your Bristol Babcock PLCs, then select **TerminalServer** as the Port Type and configure the terminal server itself with the serial port settings in the **bbhdw** output *ADD CHNcc* line. See the section above titled "Serial port properties."

Property	Description
Terminal Server TCP Host Name Terminal Server TCP Port No	The name and port number of terminal server to which the channel is connected. You can specify either a TCP host name or an IP address, but it must match the TCP host name used when you installed and internally configured the terminal server.
Idle Timeout	The time, in seconds, the channel waits for a successful connection to the server before closing the connection. A value of *0* indicates that the connection is never closed.
Checksum	The type of checksum error detection used for the port. Not applicable for this channel. Select *NONE*.
Protocol	The protocol to be used to communicate on this Bristol Babcock channel. All controllers on the channel must be configured for this protocol. The default is *BSAP*.

Main properties for a Bristol Babcock controller

Use the Main tab to define the basic properties for a Bristol Babcock controller. The controller settings must match the output of the **bbhdw** file. For example, if the output of **bbhdw** generates an *DEF RTUrrr* line as:

```
DEF RTU01.001 NODENAME=TEST NAME=TEST SIG ID=001/1 NWID=1024 MARG=25 FAIL=50
```

you would enter:

- *SIG* for Controller Type
- *001* for Local Address
- *1* for Network Level, and
- *1024* for Global Address

For information about how to create a controller, see the topic titled "Building controllers and channels" in the *Quick Builder User's Guide*.

Property	Description
Name	The unique name of the controller. A maximum of *10* alphanumeric characters (no spaces or double quotes). Note: In Station displays, underscore characters (_) appear as spaces.
Description	(Optional) A description of the controller. A maximum of *132* alphanumeric characters, including spaces.
Associated Asset	The Tag Name of the Asset to be associated with the alarm group.
Channel Name	The name of the channel on which the controller communicates with the server. (You must have already defined a channel for it to appear in this list.)
Marginal Alarm Limit	The communications alarm marginal limit at which the controller is declared to be marginal. When this limit is reached, a high priority alarm is generated. To change the priority of the alarm system wide, see the topic titled "Configuring system alarm priorities" in the *Server and Client Configuration Guide*. To change the priority of the alarm for one controller, see the topic titled "About configuring custom system alarm priorities for an individual channel or controller" in the *Server and Client Configuration Guide*. A controller barometer monitors the total number of requests and the number of times the controller did not respond or response was incorrect. The barometer increments by two or more, depending on the error, and decrements for each good call. The default value is *25*.
Fail Alarm Limit	The communications alarm fail limit at which the controller is declared to have failed. When this barometer limit is reached, an urgent alarm is generated. To change the priority of the alarm system wide, see the topic titled "Configuring system alarm priorities" in the *Server and Client Configuration Guide*. To change the priority of the alarm for one controller, see the topic titled "About configuring custom system alarm priorities for an individual channel or controller" in the *Server and Client Configuration Guide*. Set this to double the value specified for the controller Marginal Alarm Limit. The default is *50*.
Dynamic Scanning Fastest Scan Period	Select the **Dynamic Scanning** check box to enable dynamic scanning of all point parameters on this controller. The default setting for this check box is selected. Define the fastest possible scan period (in seconds) that dynamic scanning will scan point parameters on this controller. The default is *15* seconds. The dynamic scanning period does not affect the static scanning rate for a parameter. For example, if the scanning rate for a parameter is 10 seconds, and the dynamic scanning rate for the controller is 15 seconds, the parameter will still be scanned at a period of 10 seconds.

Property	Description
Controller Type	Select the controller type as one of: • SignalsOnly • ArraysOnly See the section below titled "Bristol Babcock controller types" for more information.
Local Address	*ID=addr/level* Local address in network (1 to 255). This is set via DIP switches on the device. This is the *addr* part of the ID=*addr/level* value.
Network Level	ID=addr/level Level in the network (1 to 6). This is the *level* part of the *ID=addr/level* value.
Global Address	NWID= Network Global address (decimal)
Transaction Timeout	TNS_TIME Transaction table timeout in seconds. Default is (60 * network level). This timeout value is the length of time that the server will wait for an RDB response. It should be set to the worst case RDB round trip time between the server and the controller. This will be affected by poll periods of intermediate nodes, baud, and so on.
Node Name	NODENAME= Four character node name assigned via the **aic5** software program.
Additional definitions for signal (*SIG*) controller type	The following parameters all relate to Report By Exception parameters. They correspond exactly to terminals of the RBE module. See the *ACCOL Reference Manual* for more information. **RDB Security level** (RDBSECLVL=) RDB security level Default setting is *15*. Not normally changed. **RBE Scan Rate** (RBE_PERIOD=) RBE scan period in tenths of seconds. Default is *300*. **RBE Scan Slice** (RBE_SLICE=) RBE scan slice parameter. Default is *1*. **RBE Stop Xmit** (RBE_STOPXMIT=) RBE stop transmit count. Default is *10*. **RBE Timeout** (RBE_WAITINIT=) RBE timeout between successive wait_init messages in tenths of seconds. Default is *600*. **RBE Inhibit** (RBE_INHIBIT=) Check to Inhibit use of Bristol Babcock report by exception messages. **Alarm Inhibit** (ALM_INHIBIT=) Check to Inhibit use of Bristol Babcock alarm messages. **SOE Inhibit** (SOE_INHIBIT=) Check to Inhibit insertion of Bristol Babcock alarms into the Sequence of Events file.
Item Type	The type of controller specified when this item was created.
Last Modified	The date and time the controller properties were modified.
Last Downloaded	The date and time the controller was last downloaded to the server.
Item Number	The unique item number currently assigned to this controller, in the format *RTUnnnnn*. You can change the item number if you need to match your current server database configuration. The number must be between *01* and the maximum number of controllers allowed for your system. For more information about setting the maximum value, see the topic titled "Adjusting sizing of non-licensed items" in the *Supplementary Installation Tasks Guide*.

Bristol Babcock controller types

Data within a Bristol Babcock device is divided into two types—signals and arrays. Separate controllers must be defined in the server to address signal data and array data.

Each controller type has addressing limitations as indicated:

Controller type	Server addressable range
Signals	Up to 1,927 signals per server controller
Arrays	Up to 5,461 array elements per server controller

If this array addressing capability is not adequate, extra server controllers can be configured to access the array data. There is a maximum of one server signal controller per physical Bristol Babcock controller.

Using ACCOL load files to define scanning and control strategies

After you have initial communication with the server established, you can use the ACCOL software on the server to create and download the load files. Load files define the control strategy for the node and hold all data gathered in the form of signals or arrays.

Understanding load files

Load files are created using ACCOL tools on the server and are downloaded to the Bristol Babcock node under the server's control. They are also used by the **bbpnt** utility to create a *.pnt* file for importing into Quick Builder. The process used in creating a load file is shown in "Figure 5: Producing a load file".

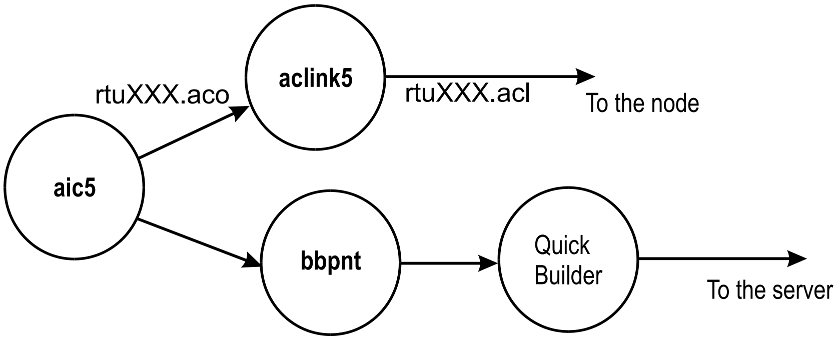

Figure 5: Producing a load file

Downloading ACL files to the Bristol Babcock node

To download the ACL load file from the server, the file must be named as follows:

```
RTUrrr.ACL
```

where *rrr* is the server RTU (controller) number for the Bristol Babcock node.

This file must exist in *server\data\bb\chnccc*, where *ccc* is the channel number of the Bristol Babcock network. Note that there will be a single load file for each Bristol Babcock node. However, because the server can have more than one controller on a particular node, there may be more than one controller in the corresponding load file. Ensure that the naming convention on your *.aco* files reflects this.

CAUTION

When downloading to controller nodes that are masters in the Network 3000 topology, all subordinate controllers must also be placed "out of service." If not, the download will be very slow and may fail. This applies equally to multiple logical controllers that reference the same Network 3000 node.

To download the .acl file

1. Put the channel "in service" and the controller "out of service." This is done on the appropriate Station displays. Download must also be permitted by the server for both the channel and the controller.
2. Initiate the download from the SCADA Controller display in Station by clicking **Download**.
3. A prompt for confirmation is displayed. Respond by typing `y` and press ENTER to continue the download.

An ACL download can take considerable time depending on the Bristol Babcock network architecture and the size of the *.acl* file. The download process reports the file record being downloaded in the Message Zone on Station.

Defining report by exception in a load

When Reporting By Exception is required it is important to remember that extended memory must be specified in the load file and the RBE module must be included.

Related topics

"Planning considerations for installing and configuring Bristol Babcock controllers" on page 5

Optimizing Bristol Babcock scanning performance

All point parameters that reference Bristol Babcock data should be assigned a scan period and be on periodic scanning. This scan period can be quite long (for example, 300 seconds).

Take into account network structure when assigning scan periods. The network restrains scanning frequency, especially to remote nodes, and aspects such as baud, poll period, and network level must be taken into consideration. For more information, see the *Bristol Babcock Network 3000 Communications User Guide*.

Periodic scanning uses the Remote Database (RDB) mechanism for retrieving data.

Scan rates for auxiliary parameters must be no faster than 60 seconds.

Reporting by Exception

If the node supports this feature it can be used to reduce traffic on the network.

Signals can be configured on RBE to report when the value changes by a specified deadband value. The RBE messages have the same priority as Remote Database (RDB) messages. Therefore, any critical signals should be configured to alarm. Alarms have the highest priority.

Attention

Avoid configuring status signals to use the alarm system as well as the RBE system. This is unnecessary duplication.

Understanding Polling

To receive data from a Bristol Babcock node it must be polled. This makes the frequency of polling very significant in terms of speed of data movement around a Bristol Babcock network.

The server will, by default, poll all top-level controllers every 2 seconds. This polling is complemented by:

- An additional poll that is issued for each acquisition request (that is, if a response is expected, a poll is also issued).
- An immediate poll if NAK is received from the node. (NAK means that the node needs polling.)
- Continuous polling (up to a limit; default 8) while data remains in the node.

Related topics

“Main properties for a Bristol Babcock channel” on page 18

Bristol Babcock scan packets

Four types of scan packet are built for an Bristol Babcock interface:

Scan packet type	Description
Hardware Diagnostic	One scan per controller every 60 seconds to verify communications integrity with the controller.
Exception Poll	Controllers are polled at the poll period.
Periodic Data Acquisition Scan	One per scan packet (arrays and RTU signal lists only).
Periodic List Data Acquisition Scan	One per scan packet (signals only).

A Bristol Babcock scan packet may consist of up to:

- 38 signals, or
- 59 array addresses

To be in a scan packet, points must reside:

For	
Arrays	• In the one server controller • Have the same scan period • Data must reside in consecutive elements of the array (in rows)
Signals	• In the one server controller • Have the same scan period

The scan packets that have been built can be listed by using the utility **lisscn** (list scan). Listing scan packets helps verify the scanning strategy.

For more information about **lisscn**, see the section titled "Command Reference" in the *Server and Client Configuration Guide*.

Defining report by exception parameters

If Report By Exception is being used for any controller then the initialization data must be specified in Quick Builder. These parameters are always downloaded to the node, and will always override any parameters set when the load was built.

CAUTION

Be careful not to set RBE_PERIOD too low initially as the communications system may be flooded with reports if there are rapidly changing signals on RBE. Leave at the default value until the network is stable.

Bristol Babcock points reference

This section describes how to configure points for a Bristol Babcock controller using Quick Builder.

In addition to the information contained in this reference, and for help to build points, see the section titled "Building and configuring points" in the *Quick Builder User's Guide*.

Related topics

"Defining a Bristol Babcock address for a point parameter" on page 30
"Defining a Bristol Babcock address for points that reference signals" on page 31
"Defining a Bristol Babcock address for points that reference arrays" on page 34

Defining a Bristol Babcock address for a point parameter

The steps for defining points with parameters that address Bristol Babcock controllers are:

Step	Done
For points that reference signals: • Using the **bbpnt** utility, create a point source file from the *aco* file. The *aco* file is produced by the Bristol Babcock **aic5** program and is a complete definition of the load that resides in a particular controller. See the topic "Defining a Bristol Babcock address for points that reference signals" for more information. • Import the **bbpnt** output file into Quick Builder. • Configure alarms for these points as required. See the topic titled “Configuring alarms for signal data” for more information.	
For points that reference arrays, you define these points wholly within Quick Builder. (The **bbpnt** utility does not generate a point definition file for points that reference arrays.) See the topic titled "Defining a Bristol Babcock address for points that reference arrays" for more information.	
When you have finished defining or editing both types of points in Quick Builder and are ready to download them, disable the Bristol Babcock channel on the server.	
Download the point definitions to the server.	
Enable the Bristol Babcock channel.	

Related topics

“Planning considerations for installing and configuring Bristol Babcock controllers” on page 5
“Defining a Bristol Babcock address for points that reference signals” on page 31
“Configuring alarms for signal data” on page 33
“Defining a Bristol Babcock address for points that reference arrays” on page 34
“Testing Bristol Babcock communications with the server” on page 39

Defining a Bristol Babcock address for points that reference signals

Attention

You must use Quick Builder to define points that reference arrays.

The **bbpnt** utility converts a Bristol Babcock *aco* file to a server point build file. This utility generates points that address signals in the load.

Before assigning a point reference number (that is, ANA*nnn*, STA*nnn*), **bbpnt** checks to see if the point name already exists in the server database. If it does exist, the same point reference number is used. If it does not exist, the next unused point reference number is used.

It is recommended that you maintain a strict naming convention for *aco* and point source files.

CAUTION

Do not run **bbpnt** if point building is already in progress on the server.

The syntax for this utility is:

```
bbpnt acofile [options]
```

where *acofile* is the *ACCOL.aco* file from which the points will be extracted. The *.aco* extension is not required.

Option	Description
-o pntfile	Output point source file. The default is *acofile.pnt*. The *.pnt* extension does not need to be specified.
-l	Ignore local/global flag and extract all *ACCOL* signals from the *ACO* file.
-r rtunum	Controller number to which the points are assigned. The default is *1*.
-m string	Mask string. String containing up to twenty 0s and 1s indicating which positions in the signal name should be copied to the point name. Positions not specified in the mask string will be included in point names. The default will be all characters in signal name to be included up to the maximum point name length.
-s hrnum	Starting character position in the signal name as stored in the ACO file for the point name. The 16-character point name is extracted from the signal name starting at this position. Defaults to the first character position, *1*.
-a code	Area code. A two-character code specifying the area to which the points will be added. The default is no area code.
-p pntnum	Point number. Starting point number for all types of points. The default is *1*.
-v	Verbose. Print out extra information for debugging.

Examples

`bbpnt load3`	Extract all global signals from *load3.aco* and put them in *load3.pnt*.
`bbpnt load3 -l`	Extract all signals (local and global).
`bbpnt load3 -r 3`	Assign all points to controller number 3.
`bbpnt dpc.aco -r 3 -a n1 -p 500 -o rtu003.pnt`	Extract global signals from controller number 3 from *dpc.aco* and put them in *rtu003.pnt*. Set the area of all points to N1 and start point numbering at 500.

Applicable analog signals in the *aco* file will produce output for the point definition file. What is produced in the point definition file depends on the contents of the *aco* file. For example, *A1SOURCE* and *A1DESTIN* entries are only produced for analog alarm signals that have one of their alarm limits assigned to another signal. As much relevant information as can be extracted from the *aco* file will be put into the point definition file.

Example

```
DEL Point_name
ADD Point_name ANAnnn analog
RANGE Point_name 0 100.0
PVSOURCE Point_name rrr signal_name
OPSOURCE Point_name rrr signal_name
OPDESTIN Point_name rrr signal_name
MDDESTIN Point_name rrr signal_name
PVPERIOD Point_name 60O
PPERIOD Point_name 60
A1SOUCE Point_name rrr signal_name
A1DESTIN Point_name rrr signal_name
A1PERIOD Point_name 60
A2SOURCE Point_name rrr signal_name
A2DESTIN Point_name rrr signal_name
A2PERIOD Point_name 60
A3SOURCE Point_name rrr signal_name
A3DESTIN Point_name rrr signal_name
A3PERIOD Point_name 60
A4SOURCE Point_name rrr signal_name
A4DESTIN Point_name rrr signal_name
A4PERIOD Point_name 60
A1NAME Point_name AL1
A2NAME Point_name AL2
A3NAME Point_name AL3
A4NAME Point_name AL4
ALMLIM1 Point_name limit 6 0
ALMLIM2 Point_name limit 7 2
ALMLIM3 Point_name limit 8 1
ALMLIM4 Point_name limit 9 3
```

Applicable logical signals in the *aco* file will produce the following output for the point definition file. The output will be in the following form.

Example

```
DEL Point_name
ADD Point_name STAnnnn status
RANGE Point_name 0.0 1.0
STATEDES Point_name OFF ON
PVSOURCE Point_name rrr signal_name
OPSOURCE Point_name rrr signal_name
OPDESTIN Point_name rrr signal_name
MDDESTIN Point_name rrr signal_name
PVPERIOD Point_name 60
OPPERIOD Point_name 60
```

Related topics

"Defining a Bristol Babcock address for a point parameter" on page 30

"Address syntax for Bristol Babcock controllers for points addressing an array" on page 34

Customizing point source files

If required, point source files generated by **bbpnt** can be customized using a text editor. The next step is to import the point build files into Quick Builder. See the *Quick Builder User's Guide* for information about importing point files.

After the points have been imported into Quick Builder, they can be downloaded to the server.

Importing the point build files into Quick Builder

Point build files generated by **bbpnt** can be imported into Quick Builder and amended. For example, you might want to change the Scan Period entries and perhaps add comments for each point or attach algorithms.

Configuring alarms for signal data

Alarms are used to update point parameters and are the quickest way of transmitting data over the network. Alarms can only be configured for points built on signal data.

Use alarms to:

- Update point parameters.
- Quickly transmit data over the network.
- Reduce periodic scanning.
- Store as events in order to retain Bristol Babcock alarm timestamp information. (The Sequence of Events (SOE) file is used for this purpose.)

Reducing periodic scanning with alarms

Place a status point parameter on a slow scan period (300 seconds) and configure the signal in the node to alarm on a change of state.

Retaining timestamp information

An event report entry includes the timestamp received in the alarm report to a resolution of one millisecond. If an alarm value is used to update a process variable, and this causes an alarm to occur, the alarm will be time stamped with the system time not the time in the alarm report. The difference between these timestamps could be significant depending on the network architecture, hence the need to retain the "real" alarm time in the SOE file.

See the *Server and Client Configuration Guide* for more information on SOE configuration and SOE reports.

Attention

- A point reference for the alarm must exist if an SOE entry is to be produced.
- Alarms must be enabled for the point if an SOE entry is to be produced.
- The “extended alarm format” (that has signal inhibit and alarm inhibit information) is not supported.

Related topics

“Defining a Bristol Babcock address for a point parameter” on page 30

Defining a Bristol Babcock address for points that reference arrays

CAUTION
Disable the Bristol Babcock channel on the server before downloading points from Quick Builder.

Related topics

“Address syntax for Bristol Babcock controllers for points addressing an array” on page 34
“Defining a Bristol Babcock address for a point parameter” on page 30

Address syntax for Bristol Babcock controllers for points addressing an array

For source and destination addresses the format for a Bristol Babcock controller address is:

```
ControllerName Address
```

Part	Description
ControllerName	The name of the Bristol Babcock controller.
Address	The address in the controller where the value is recorded. In the case of an Array Address, see the section below titled "Array address." In the case of a Signal Address, you use the **bbpnt** utility, see the topic titled "Defining a Bristol Babcock address for points that reference signals."

If you would like help with the address, you can use the Address Builder. To display the Address Builder, click ... next to **Address**.

Array address

For points referencing arrays, definitions are done completely within Quick Builder. The address syntax for a point addressing an array is:

```
ArrayNumber Row Column
```

Part	Description
ArrayNumber	The array number— must be in the range 1 to 255
Row	The row number
Column	The column number—must be in the range 1 to 255

Valid point addresses

Valid point addresses specific to the Bristol Babcock interface that can be defined for the server point types are shown. Note the difference between signal and array addressing.

Address	Controller type Signals	Controller type Arrays
Analog Point		
PVSOURCE	Yes	
SPSOURCE	Yes	
SPDESTIN	Yes	
OPSOURCE	Yes	Yes

Address	Controller type Signals	Controller type Arrays
OPDESTIN	Yes	Yes
MDSOURCE[1]		
MDDESTIN	Yes	
AxSOURCE	Yes	Yes
AxDESTIN	Yes	Yes
Status Point		
PVSOURCE	Yes	Yes
OPSOURCE	Yes	Yes
OPDESTIN	Yes	Yes
MDSOURCE[1]		
MDDESTIN	Yes	
Accumulator Point		
PVSOURCE	Yes	Yes

Related topics

"Defining a Bristol Babcock address for points that reference signals" on page 31

[1] The *MDSOURCE* entry should not be specified for a Signal address. The Bristol Babcock interface will automatically read the mode value from the signal configured as *PVSOURCE*.

Troubleshooting Bristol Babcock issues

This section describes troubleshooting tasks for Bristol Babcock that you can perform either on the server or from any Station.

Related topics

“Troubleshooting Bristol Babcock point configuration errors” on page 38
“Testing Bristol Babcock communications with the server” on page 39
“Troubleshooting Bristol Babcock scanning errors” on page 40
“Forcing initialization of point addresses” on page 41

Troubleshooting Bristol Babcock point configuration errors

When downloading to the server the errors reported specific to the Bristol Babcock are:

Error	Reason
Too many fields	Check file for unexpected field.
Controller type not supported	Check that hardware was built correctly.
Signal name is too long	Check the full signal name is less than or equal to 20 characters.
Invalid Bristol Babcock array number	Check the array number is in the range 1 to 255.
Invalid Bristol Babcock array column number	Check the column number is in the range 1 to 255.

Testing Bristol Babcock communications with the server

You use the Bristol Babcock test utility, **bbtst**, to test communications between the server and the Bristol Babcock controller after you have downloaded channel and controller definitions to the server database.

Prerequisites

- Set up the controller.
- Connect all cables.
- Define the controller and channel in Quick Builder.
- Download the Quick Builder definitions to the server, without errors.
- Ensure the channel is out of service.

To run the bbtst utility

1. Open a Command Prompt window.
2. Type **bbptst** and then press Enter.
3. Follow the directions as prompted.
   The device serial interface transmit and receive LEDs flash during transactions.

   If any errors are encountered, review the channel and controller definitions in Quick Builder and make sure they match the **bbhdw** output.

Results

After you verify that the server is communicating with the Bristol Babcock controllers you can configure points. See the topic titled "Defining a Bristol Babcock address for a point parameter."

Related topics

“Planning considerations for installing and configuring Bristol Babcock controllers” on page 5
“Defining the connections to controllers” on page 16
“Defining a Bristol Babcock address for a point parameter” on page 30

Troubleshooting Bristol Babcock scanning errors

If the Point Detail display for a point shows a bad value (indicated by inverse video), the point might built with an address that is syntactically valid but not configured in the controller.

The controller responds to a request to read this address with a NAK message. Note that NAK messages received in response to a read request by the server do not contribute to the error count of the controller or channel.

NAK messages received in response to a write request by the server result in an operator message being sent to the Station as well as contributing to the controller's error count.

Any NAK messages received are printed in the server error log (which can be viewed via the Server Diagnostics Program Group).

If A-NAK messages occur, check the relevant point address to ensure there are no references to addresses in the controller that are not actually configured within the controller.

Forcing initialization of point addresses

The server maintains addresses that correspond to signal names. When periodic scanning or Reporting by Exception is initialized, signal names are used until the addresses have been determined. From then on, all communication messages use addresses to reduce the message size.

If the load changes, it is necessary to re-initialize the signal addresses. This is done automatically, but can be forced using the **padrst** utility.

To manually force initialization of a point address

- At the Command Line, type `padrst` *nnn*, where *nnn* is the controller, and press ENTER.

Notices

Trademarks

Experion®, PlantScape®, SafeBrowse®, TotalPlant®, and TDC 3000® are registered trademarks of Honeywell International, Inc.

OneWireless™ is a trademark of Honeywell International, Inc.

Other trademarks

Microsoft and SQL Server are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Trademarks that appear in this document are used only to the benefit of the trademark owner, with no intention of trademark infringement.

Third-party licenses

This product may contain or be derived from materials, including software, of third parties. The third party materials may be subject to licenses, notices, restrictions and obligations imposed by the licensor. The licenses, notices, restrictions and obligations, if any, may be found in the materials accompanying the product, in the documents or files accompanying such third party materials, in a file named third_party_licenses on the media containing the product, or at http://www.honeywell.com/ps/thirdpartylicenses.

Documentation feedback

You can find the most up-to-date documents on the Honeywell Process Solutions support website at:

http://www.honeywellprocess.com/support

If you have comments about Honeywell Process Solutions documentation, send your feedback to:

hpsdocs@honeywell.com

Use this email address to provide feedback, or to report errors and omissions in the documentation. For immediate help with a technical problem, contact your local Honeywell Process Solutions Customer Contact Center (CCC) or Honeywell Technical Assistance Center (TAC) listed in the “Support and other contacts” section of this document.

How to report a security vulnerability

For the purpose of submission, a security vulnerability is defined as a software defect or weakness that can be exploited to reduce the operational or security capabilities of the software.

Honeywell investigates all reports of security vulnerabilities affecting Honeywell products and services.

To report a potential security vulnerability against any Honeywell product, please follow the instructions at:

https://honeywell.com/pages/vulnerabilityreporting.aspx

Submit the requested information to Honeywell using one of the following methods:

- Send an email to security@honeywell.com.

  or

- Contact your local Honeywell Process Solutions Customer Contact Center (CCC) or Honeywell Technical Assistance Center (TAC) listed in the “Support and other contacts” section of this document.

Support

For support, contact your local Honeywell Process Solutions Customer Contact Center (CCC). To find your local CCC visit the website, https://www.honeywellprocess.com/en-US/contact-us/customer-support-contacts/Pages/default.aspx.

Training classes

Honeywell holds technical training classes on Experion PKS. These classes are taught by experts in the field of process control systems. For more information about these classes, contact your Honeywell representative, or see http://www.automationcollege.com.

Index