Honeywell

Experion PKS

Peer Control Data Interface Implementation Guide

EPDOC-XX84-en-431A
February 2015

Release 431

Document	Release	Issue	Date
EPDOC-XX84-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

1 About this guide

This Guide provides information about using the licensed Peer Control Data Interface (PCDI) function to interface Honeywell's Safety Manager or third-party peer devices supporting MODBUS TCP communications with the Experion C300 Controller. It includes planning, installing, configuring, operating, and troubleshooting type data as well as some general conceptual type data to help understand the purpose of the PCDI function.

Revision history

Revision	Date	Description
A	February 2015	Initial release of the document.

Intended audience

Personnel who are responsible for interfacing Safety Manager or third-party peer devices for peer-to-peer communications through the peer control data interface with Experion control strategies.

Prerequisite skills

- Familiar with working in a Windows operating environment.
- Familiar with using these Experion applications:
  - Configuration Studio
  - Control Builder
  - Station
  - Safety Manager
  - Safety Builder
- Familiar with MODBUS TCP communications protocol.

How to use this guide

Choose a topic associated with the task you want to complete on the tree view and just click it to launch the topic in the view pane.

Related documents

The following are links to related documents for more information about associated functions.

- *Control Building Guide*
- *Control Builder Components Reference*
- *Control Builder Parameter Reference*
- *Control Builder Error Codes Reference*
- *Fault Tolerant Ethernet Overview and Implementation Guide*
- *Fault Tolerant Ethernet Installation and Service Guide*
- *Configuration Studio Overview*

- *Operator's Guide*

2 Peer Control Data Interface overview

Starting in Experion R310, the C300 Controller supports peer control data interface (PCDI) for peer device data exchange for process control. The PCDI communicates with Honeywell's Safety Manager and other Analyzers and Programmable Logic Controllers (PLCs) that support the MODBUS TCP protocol, including a serial MODBUS protocol through an off the shelf MODBUS TCP Bridge, over Honeywell's Fault Tolerant Ethernet (FTE) network. The Control Builder in Experion R310 and greater includes function blocks in its library that let you tailor the peer control data interface to meet your particular application requirements. You must purchase a PCDI license to use the related functions in Control Builder and the Experion system.

2.1 About MODBUS TCP

The MODBUS TCP is the Transmission Control Protocol/Internet Protocol (TCP/IP) version of the MODBUS protocol. It facilitates communication between devices connected on an Ethernet TCP/IP network based on a client/server model that uses the following two types of messages with standard TCP acknowledge in response to a message.

- MODBUS Request: Client sends this message on the network to initiate a transaction
- MODBUS Response : Server sends this message in response to Client

A MODBUS TCP based communication system can include the following different types of devices.

- MODBUS TCP/IP client and server devices connected to a TCP/IP network.
- Bridge, router, or gateway interconnection device to link the TCP/IP network to a serial line sub-network, which permits connections to MODBUS serial line client and server end devices.

For more information about MODBUS TCP and MODBUS protocols, visit the *MODBUS organization* web site.

2.2 Peer Control Data Integration over FTE

Honeywell's Fault Tolerant Ethernet (FTE) serves as the communications media for C300 Controllers to provide peer control data interface with Safety Manager and other peer devices.

The Safety Manager connects directly to the FTE network through Yellow and Green cables or optionally through the CF9 control firewall as shown in the following figure. However, the current Safety Manager is considered a Non-FTE node.

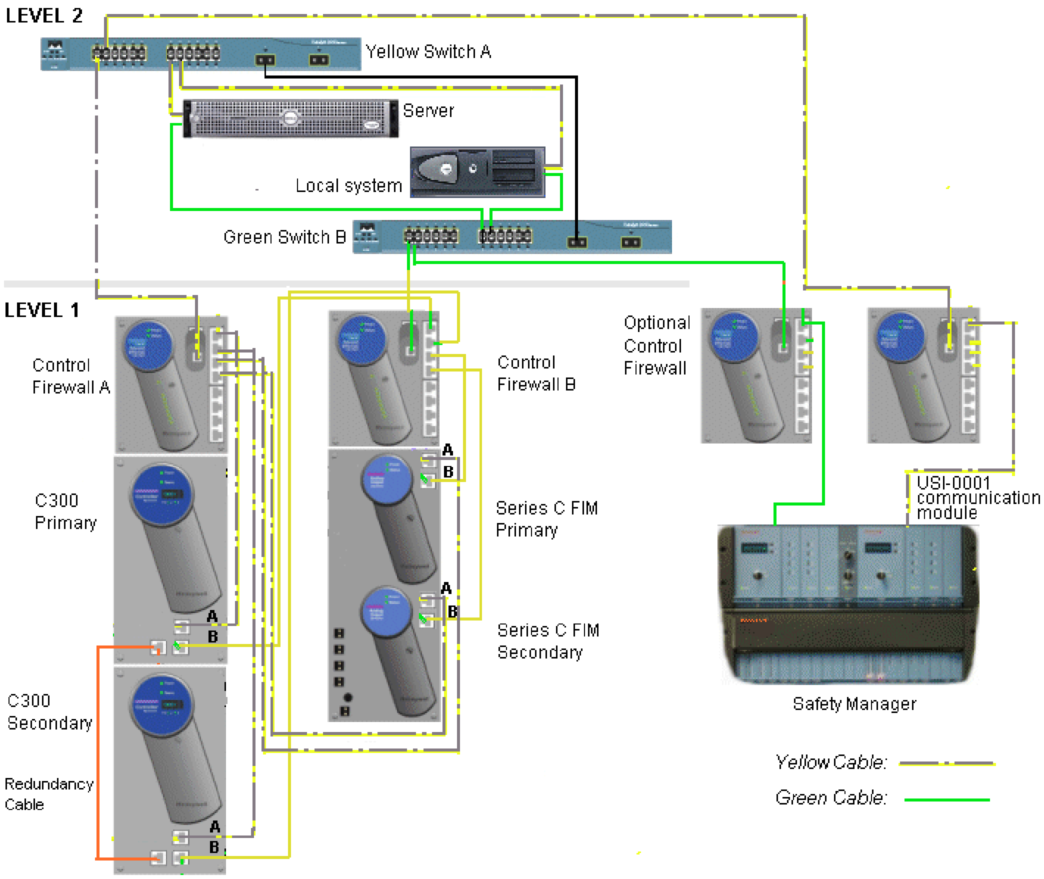


Figure 1: Typical C300 Controller Peer Control Data Interface Topology with Redundant Safety Manager

A peer device connects to either the Yellow or the Green side of the FTE network, as shown in the following figure; or optionally through the CF9 Control Firewall. For redundant devices, the common connection configuration is the Yellow side to the primary device and Green side to the secondary device. For MODBUS remote terminal unit (RTU) peer devices on a serial line connected to a MODBUS TCP peer Bridge, either FTE side connects to the bridge.

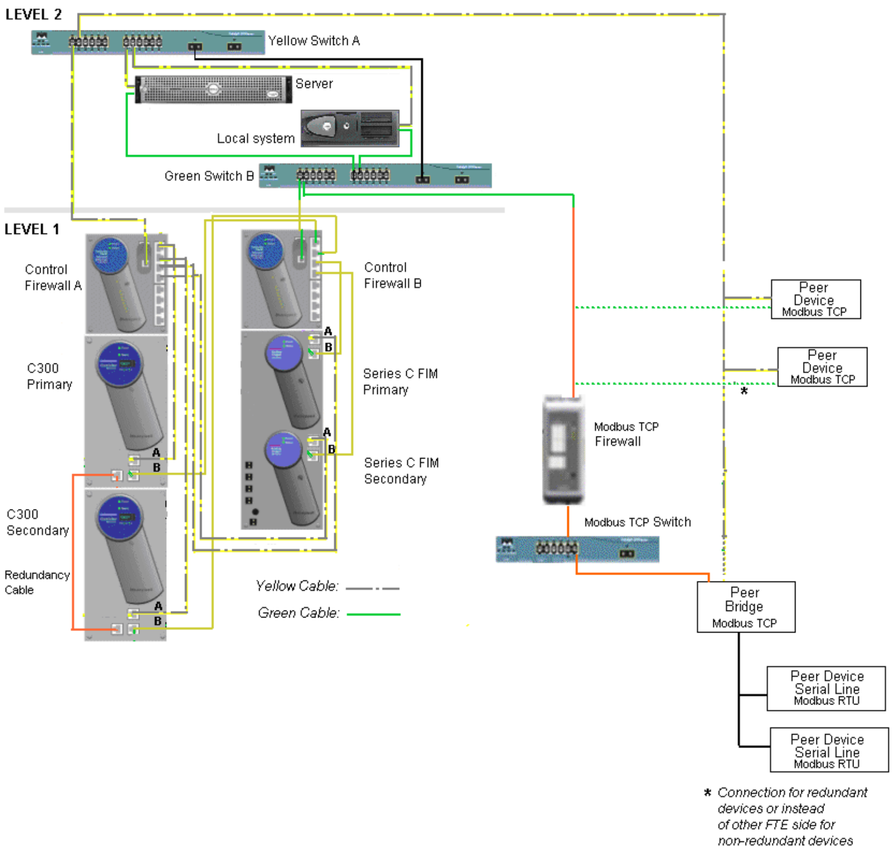


Figure 2: Typical C300 Controller Peer Control Data Interface Topology with Third-Party MODBUS TCP Devices and Bridge

2.3 PCDI Library

Beginning in Experion R310, the Library tab in the Experion Control Builder includes the following peer control data interface (PCDI) library of PCDI device and request/channel function blocks to support configuration of Safety Manager or third-party MODBUS TCP components with Experion control strategies.

⊟ PCDI Block Name and Icon	Description
PCDI_MASTER	This block represents a Safety Manager, native MODBUS TCP Device or MODBUS TCP Bridge with serial bus connected MODBUS RTU devices. It is a stand-alone block that must be assigned to a CEEC300.
PCDIFLAGARRCH	The Flag Array Request Channel block allows Boolean access to coils and discrete data access in associated Safety Manager or MODBUS TCP device. It is a basic block that must be contained in a Control Module with channels assigned to applicable PCDI device.
PCDINUMARRCH	The Numeric Array Request Channel block allows access to registers in associated Safety Manager or MODBUS TCP device. It is a basic block that must be contained in a Control Module with channels assigned to applicable PCDI device.
PCDITEXTARRCH	The Text Array Request Channel block allows access to ASCII text in associated Safety Manager or MODBUS TCP device. It is a basic block that must be contained in a Control Module with channels assigned to applicable PCDI device.

2.4 Basic Peer Control Data Interface Block Architecture

As shown in the following illustration, the PCDI_MASTER block serves as the communications bridge between PCDI Array Request Channel blocks and Safety Manager, peer device or peer bridge with serial line MODBUS RTUs. You can configure the PCDI_MASTER block to represent a single or redundant Safety Manager, single or redundant peer (MODBUS TCP) device, or single or redundant peer (MODBUS TCP) bridge. Each PCDI_MASTER block supports up to 64 PCDI Array Request Channel Blocks. Each peer bridge handles up to 16 serial MODBUS RTU connections. The actual total number of devices supported may be restricted by the system license details. See the following “Checking license details” on page 97 section for more information.

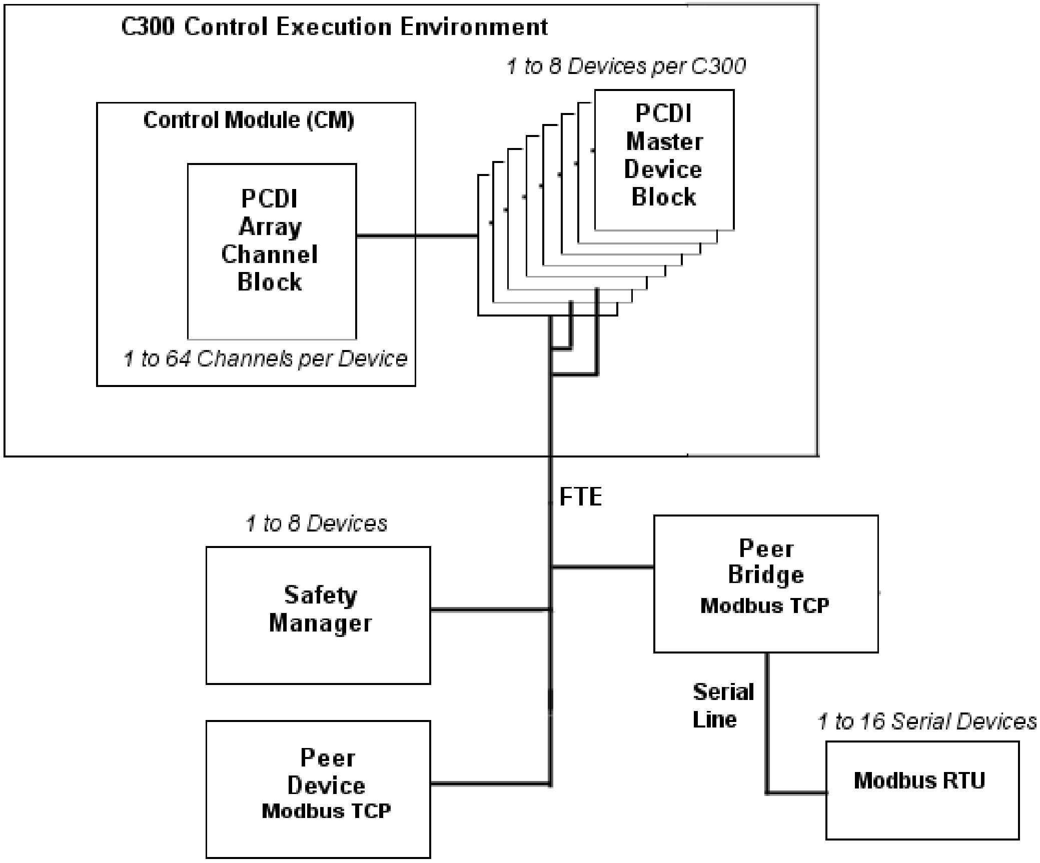


Figure 3: Simplified PCDI block architecture and signal path

2.5 Safety Manager integration

The PCDI lets you efficiently integrate Honeywell's Safety Manager with C300 Controllers.

Please refer to the *Safety Manager integration Guide* for more information about integrating Safety Manager with Experion C300 Controller.

2.6 Peer Control Data Interface processing characteristics

The following table summarizes some peer control data interface processing characteristics for the given function.

Function	Processing Characteristic
Scheduling	The PCDI_MASTER block is executed every basic cycle (50 milliseconds) prior to the execution of any Control Modules. Each PCDI_MASTER block has its own EXECSTATE. The EXECSTATE is Inactive when the block is loaded. You set the EXECSTATE to Active the same as you would a Control Module after loading.
	The PCDI Array Request Channel block is executed at the execution cycle of its container Control Module. The block does not have its own EXECSTATE. The EXECSTATE of the Control Module controls the execution state of the PCDI Array Channel block. The PCDI Array Request Channel block runs in either the triggered or the auto-triggered mode. In triggered mode, the block submits a request on the positive transition of the input trigger flag. In auto-triggered mode, the DONE flag is internally looped back to the trigger to provide for the fastest update rate possible. In this mode, the block is capable of refreshing data at the same rate as the execution cycle of the containing CM - limited by the response time of the addressed device
Alarming	The PCDI_MASTER block serves as the notification detector for all PCDI related alarms. The alarms are classified as System Diagnostic Notification ones that map to the System Alarm category with condition name of DIAG on Experion Alarm Summary, System Status Display and Event Summary on Station. The PCDI Array Request Channel blocks will report channel alarms through the PCDI_MASTER block it is attached to. If an alarm condition no longer exists or the block becomes inactive or it is deleted, channel alarms will return to normal.
Checkpointing	The PCDI_MASTER block and PCDI Array Request Channel blocks supports the standard checkpoint saves and restores functionality. See the “PCDI support for checkpoint save/ restore functions” on page 99 section for more details.
Redundancy	In addition to the redundancy capability of the C300 Controller, peer control data interface supports two connections to two devices at different IP addresses in the PCDI_MASTER block. Depending on the connection of the two MODBUS TCP devices, an additional protection against network failures is possible if one of the devices is connected to the Yellow side of FTE and the other device to the Green side.
Whole Array Access	The PCDI Array Request Channel blocks are designed to read or write array data. Array data are accessible on a single exposed pin as whole-array data, or on individual pins specified by array index. Array data is type specific and supports Flags, Numerics, or Textual data. Control Builder will only allow connecting array pins of the same types to the data pins of these function block pins. Array connections can be carried across Control Module boundaries using Named Parameter connectors for connections between Control Modules.

Function	Processing Characteristic
Read/Write	The PCDI Array Request Channel block data parameters can be exposed as either Input or Output pins on the channel blocks. The parameters can also be referenced by name without the pin being exposed. When used as input pins or referenced by parameter name, the block will 'Pull' data from the connected block, and write the data to the associated end device. When the pin is used as an output pin or referenced by parameter name, the data will be read from the end device and made available to the connected block. An operator can also modify data by overwriting values in the Monitoring view. All PCDI Array Request Channel blocks read data from the assigned address range. Blocks that perform writes will first write and then read the data. The DONE parameter will not be true until both the read and write part of the cycle are performed. Any errors during the read part of the cycle will result in a BADPV. Any errors during either the write or the read will set the ERRORCODE with the error reason. The ERRFL flag will reflect whether the error condition is still active. The ERRFL will turn off when a successful write/read cycle has completed. Since the last error message is never cleared, even after a successful read or write, users must always check the error flag first to confirm that there really is an error.
Write	Write of data to a PCDI end device is governed by one of these configurable methods. • WriteOnDiff: Compares the value to write with the PV (last value read). If they are different, a write is required. • WriteAlways: Performs a write regardless of the current value or the value being written. A read occurs after the write has been completed. • WriteOnChange: Compares the value to write with the last value written. If they are different, a write is required. • WriteAlwaysWriteOnly: Same as WriteAlways. However, a read is not performed after the write. • WriteOnChgWriteOnly: Same as WriteOnChange. However, a read is not performed after the write.
Array Read	The read of Modbus arrays configured in PCDI Array Request Channel blocks occur on the rising edge of the trigger input. If the AUTOTRIGGER configuration parameter is set, then the next read will occur in the next CM period if the done flag was returned before the beginning of the next cycle. The done flag will be tested at the beginning of the cycle and the read will be performed, if the flag is set. The maximum array sizes are all designed to be delivered in one MODBUS TCP packet. The data returned will be formatted according to the data type defined by the address range of the PCDI Array Channel block.

3 Peer Control Data Interface planning and design

Related topics

“Peer Control Data Interface requirements” on page 20
“MODBUS system considerations” on page 21
“Write option selection considerations” on page 22
“Performance considerations” on page 23
“Safety Manager performance considerations” on page 24

3.1 Peer Control Data Interface requirements

Be sure your Experion System meets the following minimum requirements to implement peer control data interface functions.

- License for peer control data interface components.
- Experion Server running R310 or greater software.
- Experion Station and Control Builder R310 or greater versions.
- Non-Redundant or Redundant C300 Controller with compatible firmware.
- SIM-C300 Controller for simulation support is optional.

Attention

The Experion C200 Controller does not support peer control data interface functions. You must upgrade to the Experion C300 Controller platform to use the peer control data interface.

3.2 MODBUS system considerations

The following are some things you should consider when planning your MODBUS system for peer control data interface.

- Record the primary and secondary IP addresses for each MODBUS TCP Device.
- Map how you want to set up the MODBUS system in relation to the 64 request channels available per PCDI_MASTER block and considering that each MODBUS TCP Bridge can have a maximum of 16 MODBUS RTU devices.
- List of MODBUS RTU devices and key properties that will be mapped to the PCDI Array Request Channel blocks.
- MODBUS RTU devices are addressed from 1 to 247 and 255. A value of 0 (zero) indicates no device is configured (removes the device). Addresses 248 to 254 are reserved.
- You can configure the PCDI_MASTER block to represent a MODBUS TCP Bridge that can support a maximum of 16 MODBUS RTU devices.

3.3 Write option selection considerations

You can select how the Array Request Channel block will handle writes to its '*data*' parameter through the configuration of the Write Option (WRITEOPT) parameter on the block's configuration form in Control Builder. The following table lists some things to consider when making the Write Option selection.

If Data Characterization Requirements Are That . . .	Then, Consider This . . .
Multiple elements change before a request block DONEFL becomes TRUE and all elements have an input pin.	The *WriteAlways* or *WriteAlwaysWriteOnly* selection will be the most efficient. This function has the possibility of transferring larger messages, but because the elements will be contiguous, they can be transferred in a single multiple write message. If the device does not support multiple element writes, this will be the most expensive function, since every element will be transferred with a separate write message.
The input data rarely changes.	The *WriteOnChange* or *WriteOnChangeWriteOnly* mode is probably a good option. If multiple element writes are supported, all contiguous elements that have changed will be transferred in single multi-element write messages. Multiple messages will be required, if there are breaks in the indexes that have changed.
The device data being read may change independent of values written, and it is desired to force the device value to the input value.	The *WriteOnDiff* selection is the best function.

3.4 Performance considerations

The following table lists some C300 and MODBUS performance related data for general planning purposes. *This information is subject to change without notice*.

C300 Controller[1]	
Number of PCDI_MASTER blocks per C300 Controller	8 maximum (One block per connected Safety Manager, peer device or peer bridge.)
Number of Serial MODBUS RTUs per C300 Controller	128 maximum (16 devices maximum per MODBUS TCP Bridge times 8 MODBUS TCP Bridges maximum per C300 Controller.)
Number of Channels per PCDI_MASTER block	64 maximum
Bytes per Second per Channel	256
Message per Second	1 (A maximum message is 256 bytes.)
Performance Estimate[2]	$64 * 1 * 8 * 8 = 4096XU$
MODBUS[3]	
Byte Rate	$ByteRate = \frac{Baudrate}{11}$
Maximum Packet Size for Reads	256 bytes for all function codes (The request packet is 6 bytes long.)
Maximum Packet Size for Writes	256 bytes (The acknowledgement packet is 6 bytes long.)
Maximum Packet Size Rate	$PacketSizeRate = \frac{Baudrate}{11} * (256 + 6)$
Rate per Request	Baudrate * 2915 + device processing time
Maximum Data Delivery	Takes into account a single Array Request Channel block at 50 milliseconds. The maximum rate would be 64 times this rate.

Notes:

1. The use of write on change or write on difference for arrays of flags or numerics will have an effect on C300 performance due to the number of requests that will be queued. The individual elements of the array that change will each generate a request. The calculation of the impact on the C300 execution units must include the maximum number of writes that will occur for array writes. To minimize the C300 impact, write always for arrays should be used. However, this will have a negative impact on serial MODBUS connections due to the write of data that is identical to what is already in the MODBUS device.
2. Assumes that performance will be similar to Exchange block requirement of 8 XUs (exchange units) per message per second.
3. The use of write on change or write on difference for arrays of flags or numerics will have an effect on performance due to the number of requests that will be queued. The individual elements of the array that change will each generate a request. The performance of the low speed MODBUS serial bus will be impacted by the write of a whole array where only some elements change.

3.5 Safety Manager performance considerations

The following table lists some Safety Manager performance related data for general planning purposes. *This information is subject to change without notice.*

Safety Manager Communications Performance	
Maximum Number of parameters that can be Read from Safety Manager by C300	Unlimited PPS [1]
Maximum Number of parameters that can be Written to Safety Manager from C300 (single writes exclusively)	64 per cycle [2]
Maximum Number of parameters that can be Written to Safety Manager from C300 (block writes exclusively)	592 bytes per cycle [2]
Bandwidth usage formulas for a mixture of single writes and block writes (time and size are the key constraints)	See equations below table
Notes: 1. Safety Manager multi-processor architecture enables support of an unlimited number of reads. 2. 'Cycle' refers to the Safety Manager application cycle.	

To avoid overruns, writes to Safety Manager must be kept within the following limits:

$$\left(\frac{\#\ \text{Write messages per second}}{2}+\frac{\#\ \text{Bytes written per second}}{250}\right)\leq 25$$

and

$$\sum_{0}^{n} \mathit{Mblock}+\sum_{0}^{m} \mathit{Rblock}+\left(10\times \mathit{Single}\right)\leq 640\text{bytes}$$

Where:

Single = a single write command

Mblock = marker or coil block write command

Rblock = register block write command

Note that the minimum number of bytes written per message is 4.

4 Peer Control Data Interface installation and upgrades

Related topics

“Experion software installation” on page 26
“FTE network installation” on page 27
“Hardware installation” on page 28
“Safety Manager software installation” on page 29
“General installation considerations and restrictions” on page 30

4.1 Experion software installation

The Peer Control Data Interface is provided as part of the Experion R310 software. Refer to the Experion *Software Installation User's Guide* for details about installing the Experion software.

4.2 FTE network installation

Refer to the *Fault Tolerant Ethernet Installation and Service Guide* for information about installing a FTE network. It is beyond the scope of this document to cover specific FTE installation details.

4.3 Hardware installation

Refer to the applicable Honeywell or vendor component documentation for details about installing any Experion control hardware or MODBUS TCP related hardware.

4.4 Safety Manager software installation

The Peer Control Data Interface support is provided as part of the Safety Manager R130 software. Refer to the *Safety Manager Installation and Upgrade Guide* for details about installing Safety Builder software.

4.5 General installation considerations and restrictions

The following table lists some general considerations and/or restrictions related to a given installation function.

Function	Consideration/Restriction
Network Connection	You may connect peer devices to Level 2 configured FTE network ports, just as you would an Experion server or console. You must configure the devices for 100 megabit full duplex operation. Avoid using bridges or end devices that can only run at 10 megabits. For redundant bridges or end devices, we recommend connecting one device to the Yellow switch side and one to the Green switch side of FTE. If devices have built in network redundancy, connect one cable to Yellow and one to Green. Be aware that the current Safety Manager (SM) is a Non-FTE node. This means that SM may demonstrate single node connection behavior in a redundant topology in response to a FTE network problem, such as a break in the crossover cable.
CF9 Control Firewall Connection	You may connect qualified peer devices or bridges to the CF9 Control Firewall in the Level 1 network. You must configure these devices for full duplex auto speed. You are not allowed to connect a computer to the Level 1 side of the Control Firewall to configure these devices.
IP Address Assignments	Follow IP address rules for either Level 1 or Level 2, depending on where the device is connected.
Serial MODBUS RTU Connections	You must connect serial MODBUS RTU devices on a serial bus, such as RS232, RS422/ RS485 full duplex, or RS485 half duplex, to the C300 MODBUS TCP subsystem through a bridge device. You must connect these devices in a daisy chain manner. A terminator may be required possibly at each end depending on the number of devices and the total length of the connecting cables.

5 Peer Control Data Interface configuration

Related topics

5.1 References

If you have never used Control Builder to build an Experion control strategy, please refer to the *Control Building User's Guide* for more information about using the application to build a control strategy.

Refer to the following documents for more details about individual parameters and function blocks.

- *Control Builder Parameter Reference*
- *Control Builder Components Reference*
- *Control Builder Error Codes Reference*

5.2 Adding Peer Control Data Interface Device (PCDI_MASTER) Block to a project

You can only view PCDI_MASTER blocks in the **Project** tab set for **Assignment** view. The **Assignment** view shows the relationship among all blocks while the **Containment** view only shows templates that contain other templates or Control Modules (CM), Sequential Control Modules (SCM), and basic blocks. To toggle the view, right-click in an open area of the tab window and select **Assignment View** or **Containment View** from the list, as applicable.

Each PCDI_MASTER block is automatically assigned a unique default tag name when it is created. If your system will include multiple PCDI_MASTER blocks, you may want to adopt a more structured syntax for naming them.

The block tag or module name must be unique to identify the block in the system. It must contain at least one letter (A-Z) and can be up to 16 characters long. Note that most special characters (such as +, \, =, /, #, etc.) and spaces are not allowed.

The item name can be up to 40 characters long and must contain at least one letter (A-Z). It is a name by which an entity is known within the context of the enterprise model. The Item name must be unique among children of the same containment parent in the Enterprise Model Builder hierarchy and should conform to the standard convention names within the system.

The module description text can be up to 132 characters long and appears on both detail and group displays.

The PCDI_MASTER block icon changes appearance in Project based upon the device type it is configured to represent.

The current Peer Control Data Interface function does not support MODBUS end device configuration.

The *Keep Alive* functionality is a TCP setting that attempts to use the same socket connection for multiple requests. Keeping socket connections in use instead of releasing them after use can result in reduced network traffic.

For redundant peer devices, the same Unit ID is applied to both the primary and secondary devices.

The Unit ID field carries the MODBUS slave address of the remote device, when addressing a device on a MODBUS+ or MODBUS serial line sub-network through a Bridge. It identifies the slave device connected on the sub-network behind the bridge. The destination IP address identifies the bridge itself and the bridge uses the MODBUS Unit ID to forward the request to the right slave device.

The PCDI_MASTER block initiates connection attempts and monitors device connection state. It establishes connections to the configured primary and secondary devices by setting the desired connection state and signaling the Requestor task when a connection activity is required. It monitors the status of pending requests for the Unit IDs configured for its MODBUS TCP device. The PCDI_MASTER controls retries of pending requests as well as times out the request once all retries have failed.

The PCDI_MASTER block monitors a timestamp value for each configured MODBUS Unit ID. The timestamp is updated whenever there is a request sent for the specific unit. If the elapsed period from the last activity to the current time exceeds the LOOPRATE parameter time, the PCDI_MASTER block submits a diagnostic request. This requires one request per serial device be allocated by the PCDI_MASTER block during initialization.

The following table summarizes how **Test Mode** (LOOPMODE) and **Data Change** (LOOPDATACHG) parameter configuration combine to determine how data is evaluated.

If Test Mode$_1$ (LOOPMODE) Is . . .	And, Data Change$_2$ (LOOPDATACHG) Is . . .	Then, Data Evaluation . . .
ReadOnly	*No Change*	Verifies that returned data value matches the Data value specified.
	Any Value	Checks for a successful read but does not verify the content.

If Test Mode[1] (LOOPMODE) Is . . .	And, Data Change[2] (LOOPDATACHG) Is . . .	Then, Data Evaluation . . .
	Increment or Invert	For Coil or Discrete Input: Same as Any Value, since it would be impossible to synchronize reads with a changing target device value. For Holding and Input Registers: Looks for a value different from the last read value.
Write/Read	*No Change*	Always writes the same value and expects the same value back from the read.
	Any Value	Only validates that a successful read occurred.
	Increment	Increments the value to write and then check for the result to read.
	Invert	Inverts the last value written and then checks for the result.

[1] For Type (LOOPTYPE) Diag Loopback Test Mode does not matter. It sends a 08/00 (diagnostic - return query data) command and checks the echoed result. For Types *Comm Event Log, Exception Status*, and *Diag Register*, Test Mode does not matter. They are always Read Only. The *Comm Event Log* and *Exception Status* Types will format the resulting data into a report string, which will be displayed in the Slave Device Definitions grid, Unit ID, Data field for MODBUS RTU devices or in the Primary Data/Secondary Data fields for native devices.

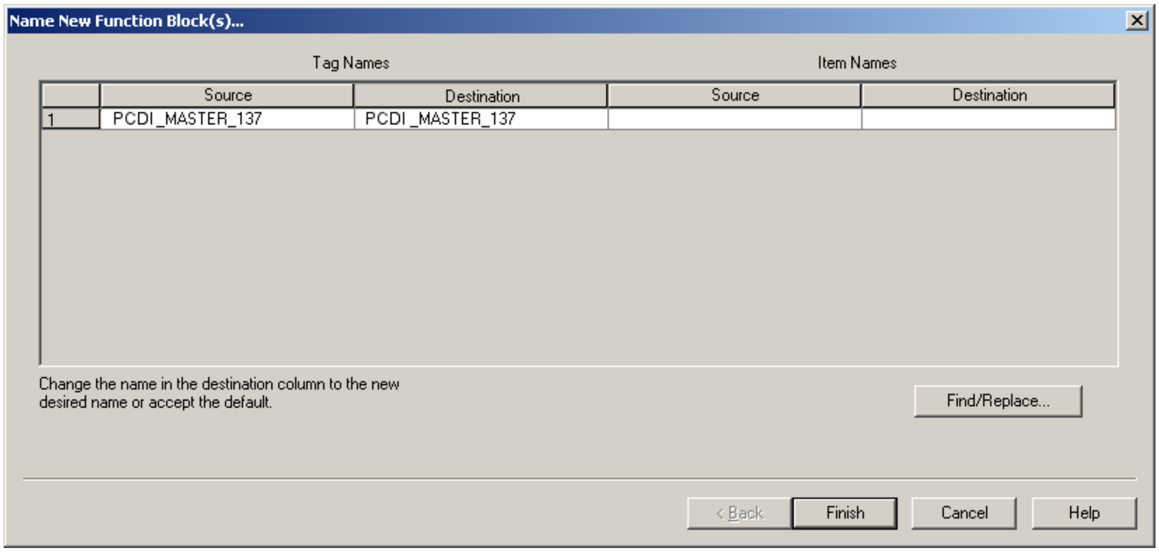
[2] The *Diag Register* Type can be queried and checked using the Data Change and Data value. Safety Manager has integrated support for this feature. Test Mode, Data, and Data Change options have no affect for Safety Manager when the Type selected is *Diag Register.*

Prerequisites

- Your Experion server is running R310 or greater software.
- You have started Configuration Studio and launched the Control Builder application.
- You have logged on with sufficient privileges to create control strategies using Control Builder.
- If applicable, you have configured the applicable IP addresses when you set up your FTE network.

To add PCDI_MASTER block to a project

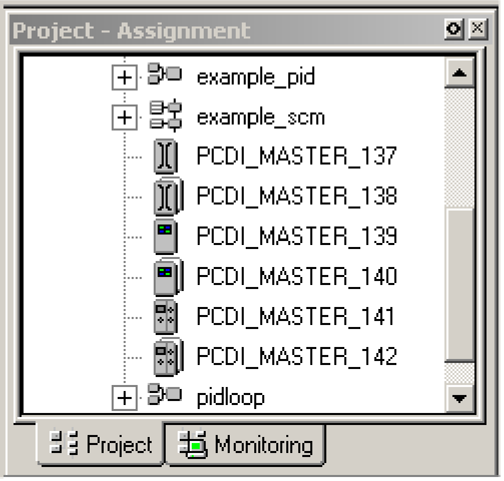
1. On the **Library** tab, click the plus sign for the PCDI library to expand the tree. Drag and drop the PCDI_MASTER block icon from the library to the **Project** tab.
2. In the open **Name New Function Block(s)** dialog, you can either:
   - Key in a new Tag Name in its **Destination** box and/or a new Item Name in its **Destination** box and click the **Finish** button to close the dialog and save the name changes. Or,
   - Click the **Finish** button to close the dialog and accept the default Tag Name and blank Item Name entries.



	Tag Names		Item Names	
	Source	Destination	Source	Destination
1	PCDI_MASTER_137	PCDI _MASTER_137		

Change the name in the destination column to the new desired name or accept the default.

3 Check that the named PCDI_MASTER block icon is now present in the **Unassigned** directory in the **Project** tab.



4 If you want to assign the PCDI_MASTER block to a C300 execution environment (CEEC300) now before configuring it, go to the following “Device supported commands” on page 43 section before continuing.

5 Go to the next section *To configure PCDI_MASTER block in Project to* continue with this procedure.

To configure PCDI_MASTER block in a project

1 Right-click the named PCDI_MASTER block icon and select **Module Properties** from the list to call up the block's Properties form for configuration.

- You can skip Steps 1 and 2, if you have already entered the desired Module Name and Item Name for the device.

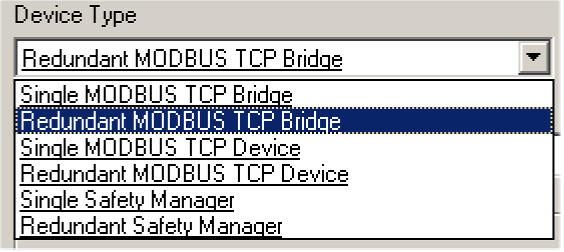
2 If you want to change the default module name, double-click the **Module Name** box to highlight it and key in the desired module name. Click the cursor in the **Item Name** box. Otherwise, go to Step 3.

3 With cursor in the **Item Name** box, key in the name of the item this object will be associated with in the Enterprise Model Builder hierarchy.

Attention

Be aware that changing the Device Type selection can clear any previously entered configuration data, since device related defaults are reset.

4 Click the down-arrow in the **Device Type** (**DEVTYPE**) box and select the Safety Manager or peer device from the list that this PCDI_MASTER block is to represent.



Device Type

Redundant MODBUS TCP Bridge

5 The **Currently Assigned Channels** list box is read only. It is automatically populated when a PCDI Array Request block is assigned to it.

6 Click the **Module Configuration** tab to display it.

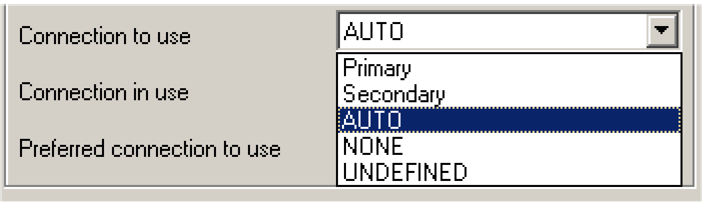
7 With cursor in **Primary IP Address** (**PRIMIP**) box, key in the IP address for the primary or non-redundant device this block represents.

8 Click the <**Tab**> key to move the cursor to the **Primary TCP Port** (**PRIMTCP**) box. All MODBUS/TCP application data unit (ADU) are sent through TCP to registered port 502. The default value is 502.

9 If you have selected a single (non-redundant)device type, you can skip this step.With cursor in **Secondary IP Address** (**SECIP**) box, key in the IP address for the redundant device this block represents. *This address must be different than one used for the primary IP address (PRIMIP)*. (Only available when redundant device type is selected.)

10 If you have selected a single (non-redundant) device type, you can skip this step.Click the <**Tab**> key to move the cursor to the **Secondary TCP Port** (**SECTCP**) box. The default value is 502. (Only available when redundant device type is selected.) An error message appears, if an invalid number, such as zero (0), is entered. A zero value cannot be stored to the database.

11 If you have selected a single (non-redundant) device type, you can skip this step.Click the down-arrow button in the **Connection to use** (**CONTOUSE**) box and select the desired connection to use. The default selection is **AUTO**, so either the primary or secondary connection is automatically selected upon a device failover.



Connection to use: AUTO

Connection in use

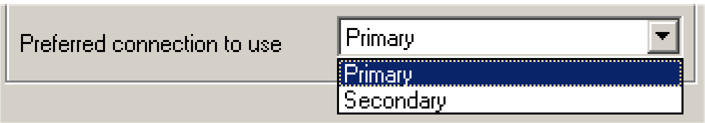
Preferred connection to use

Tip

In the Monitoring mode, you can use the CONTOUSE selection to force a specific connection. For example, you can select Secondary and it will be the only connection used while the primary is disconnected and disabled in this mode. This can be useful when performing maintenance on network or end-device equipment. An event is issued to indicate that the alternate connection is disabled. No other channel or device failures are reported, and active events at the time of the change become inactive.

12 The **Connection in use** (**CONINUSE**) box is read only and only applies for redundant devices.

13 If you have selected a single (non-redundant) device type, you can skip this step.Click the down-arrow button in the **Preferred connection to use** (**PREFERREDCONN**) box and select the preferred connection to use when the **CONTOUSE** parameter is set to Auto. The default selection is **Primary**.



Preferred connection to use: Primary

14 If you have selected a single (non-redundant) device type, you can skip this step.With cursor in the **Connection Switch Period (sec)** (**REDSWITCHPERIOD**) box, key in the desired switch time in seconds.

15 Check the **Use Keep Alive** check box to enable the option or clear the check box to disable the option. The default selection is checked or enabled.

Tip

- If Device Type setting is Single or Redundant Safety Manager, all Slave Configuration parameters are set automatically to the optimal settings.
- Users with Engineer access level or above can change the Slave Configuration parameter values, except for UNITID and supported commands, in the Monitoring mode.

16 Click the **Slave Configuration** tab to display it.

With cursor in **Timeout (ms) (DEFTIMOUT)** box, key in desired MODBUS request timeout period in milliseconds. The default is 1000 milliseconds for a native device or bridge, or 1500 milliseconds for a Safety Manager.

This value is used for all serial devices unless overridden by a serial device. The DEFTIMOUT and TIMOUT values must be greater than the configured slave timeout. This will ensure that a master block timeout does not occur when a single serial device is unresponsive.

17 With cursor in **Max Request Retries** (**REQRTRY**) box, key in desired number of retries from 0 to 10. The default is 1.

The number of retry counts the MODBUS device driver attempts after a failed read or write. Each failure is recorded. A value of zero automatically reports a timeout with no retries. This value is used for all serial devices, unless overridden by a serial device.

18 With cursor in **Per Slave Requests** (**MAXUIDREQ**), key in the desired maximum number of active requests to a serial device from 1 to 64. The default is 1.

The maximum number of active requests to a serial device. An alarm is generated if a device exceeds the active request per serial device.

19 With cursor in **Max Transactions** (**MAXPENDREQ**) box, key in the desired maximum number of transactions for a PCDI_MASTER block between 1 to 64. The default is 1 or 2 for redundant device/Safety Manager.

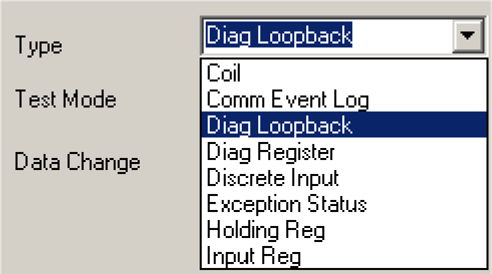
Once the maximum amount of transactions has been exceeded, an alarm is generated upon the next transaction. Transactions that exceed the maximum number of transactions will be ignored.

20 With cursor in **Message Delay (ms)** (**MSGDELAY**) box, key in the desired delay period in milliseconds. The default is 0.

(The Message Delay is the period between the last response received from the target device and the next request sent to the device.)

21 With cursor in **Rate (ms)** (**LOOPRATE**) box, key in the desired period between testing each end device in milliseconds. The default is 1000 milliseconds for native MODBUS TCP devices or 500 milliseconds for Safety Manager.

22 Click the down arrow in the **Type** (**LOOPTYPE**) box and select desired function from the list. The default is Diag Loopback or Diag Register for Safety Manager.



- If your **LOOPTYPE** selection is *Coil, Discrete Input, Holding Reg*, or *Input Reg*, go to Step 22.
- If your **LOOPTYPE** selection is *Diag Loopback*, go to Step 23.

- If your **LOOPTYPE** selection is *Diag Register*, go to Step 24.
- If your **LOOPTYPE** selection is *Comm Event Log* or *Exception Status*, go to Step 25.

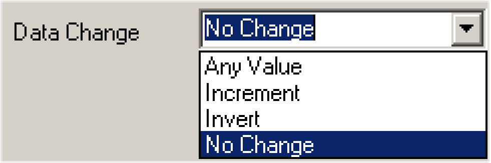
23 The **Address** (**LOOPADDR**) box is read only in **Monitoring** mode.

(This box is only available if **LOOPTYPE** selection is *Coil, Discrete Input, Holding Reg*, or *Input Reg*.)

24 With cursor in **Data** (**LOOPDATA**) box, key in desired loop data value as TRUE, FALSE, ON, OFF, 0, or 1 for *Coil* or *Discrete Input*; or as integer or hexadecimal, with prefix '0x', format for *Holding Reg or Input Reg*. The default is 0xA5A5.

(This box is only available if **LOOPTYPE** selection is *Coil, Discrete Input, Holding Reg, Input Reg*, or *Diag Loopback*.)

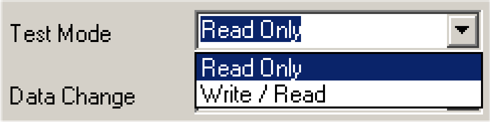
25 Click the down arrow in the **Data Change** (**LOOPDATACHG**) box and select desired function from the list. The default is No Change.



Data Change: No Change

(This box is only available if **LOOPTYPE** selection is *Coil, Discrete Input, Holding Reg, Input Reg, Diag Loopback* or *.Diag Register*.)

26 Click the down arrow in the **Test Mode** (**LOOPMODE**) box and select desired function from the list. The default is Read Only.

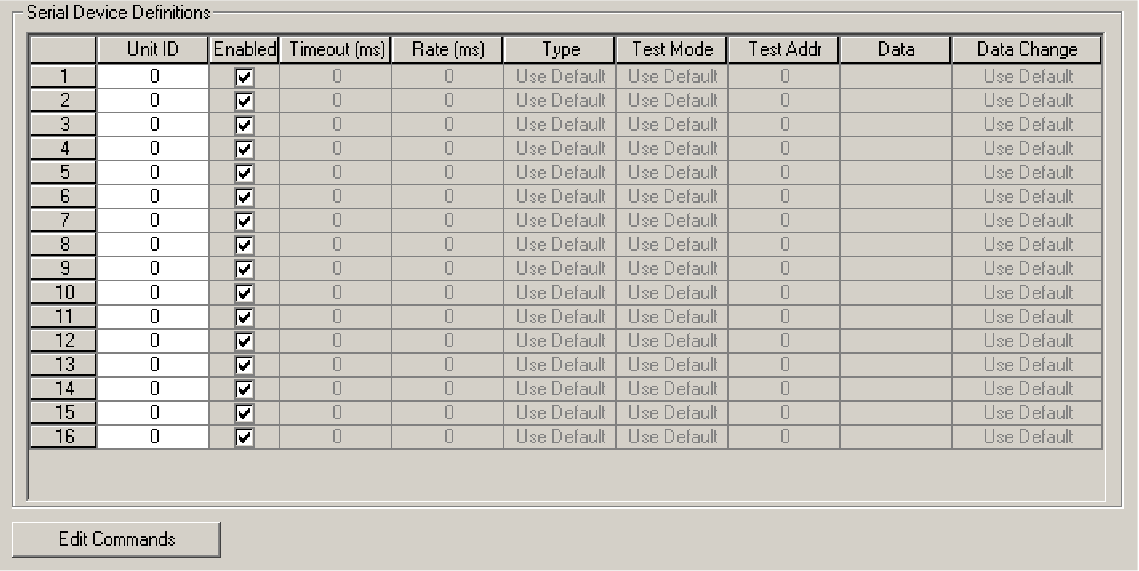


Test Mode: Read Only

- If the selected **Device Type** (**DEVTYPE**) is single or redundant MODBUS TCP Bridge, go to Step 28.
- If the Selected **Device Type** (**DEVTYPE**) is single or redundant MODBUS TCP Device, go to Step 29.
- If the selected **Device Type** (**DEVTYPE**) is single or redundant Safety Manager, go to Step 30.

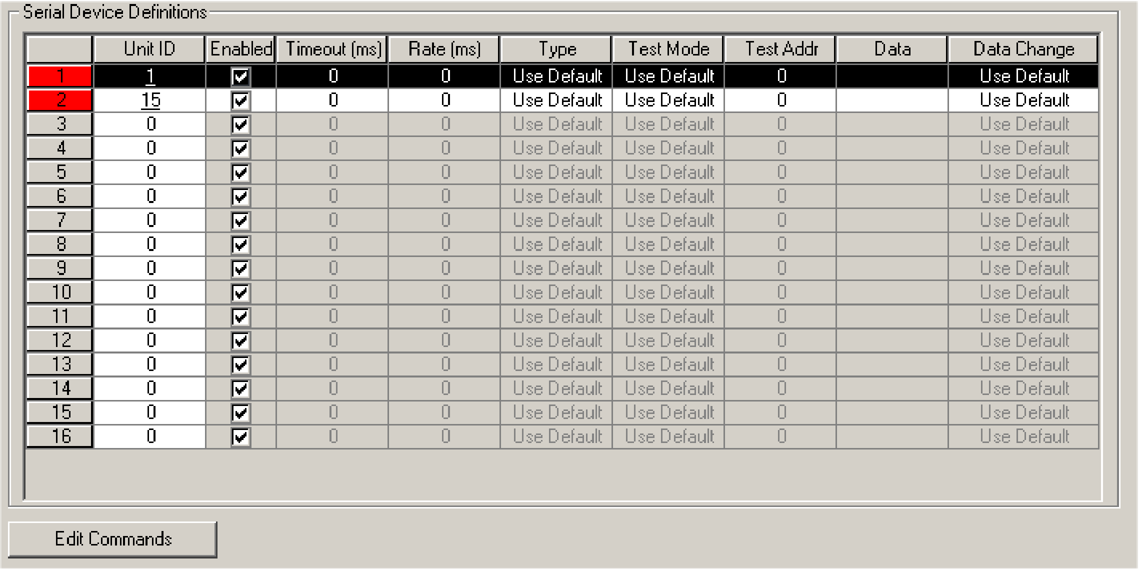
27 The **Slave Configuration** tab for a MODBUS TCP Bridge device type includes the **Serial Device Definitions** grid for configuring details about each serial device connected to the bridge.

- You must enter a valid Unit ID (1 to 247 or 255) to activate the fields in the grid for the assigned unit. A Unit ID of zero (0) is valid to remove a device, but at least one device (non-zero) must be configured.



	Unit ID	Enabled	Timeout (ms)	Rate (ms)	Type	Test Mode	Test Addr	Data	Data Change
1	0	☑	0	0	Use Default	Use Default	0		Use Default
2	0	☑	0	0	Use Default	Use Default	0		Use Default
3	0	☑	0	0	Use Default	Use Default	0		Use Default
4	0	☑	0	0	Use Default	Use Default	0		Use Default
5	0	☑	0	0	Use Default	Use Default	0		Use Default
6	0	☑	0	0	Use Default	Use Default	0		Use Default
7	0	☑	0	0	Use Default	Use Default	0		Use Default
8	0	☑	0	0	Use Default	Use Default	0		Use Default
9	0	☑	0	0	Use Default	Use Default	0		Use Default
10	0	☑	0	0	Use Default	Use Default	0		Use Default
11	0	☑	0	0	Use Default	Use Default	0		Use Default
12	0	☑	0	0	Use Default	Use Default	0		Use Default
13	0	☑	0	0	Use Default	Use Default	0		Use Default
14	0	☑	0	0	Use Default	Use Default	0		Use Default
15	0	☑	0	0	Use Default	Use Default	0		Use Default
16	0	☑	0	0	Use Default	Use Default	0		Use Default

- Click the cursor in the appropriate grid field to make entries and activate down arrow buttons to make selections, as applicable. A RED cell indicates that the Unit ID has no configured commands and is in error.

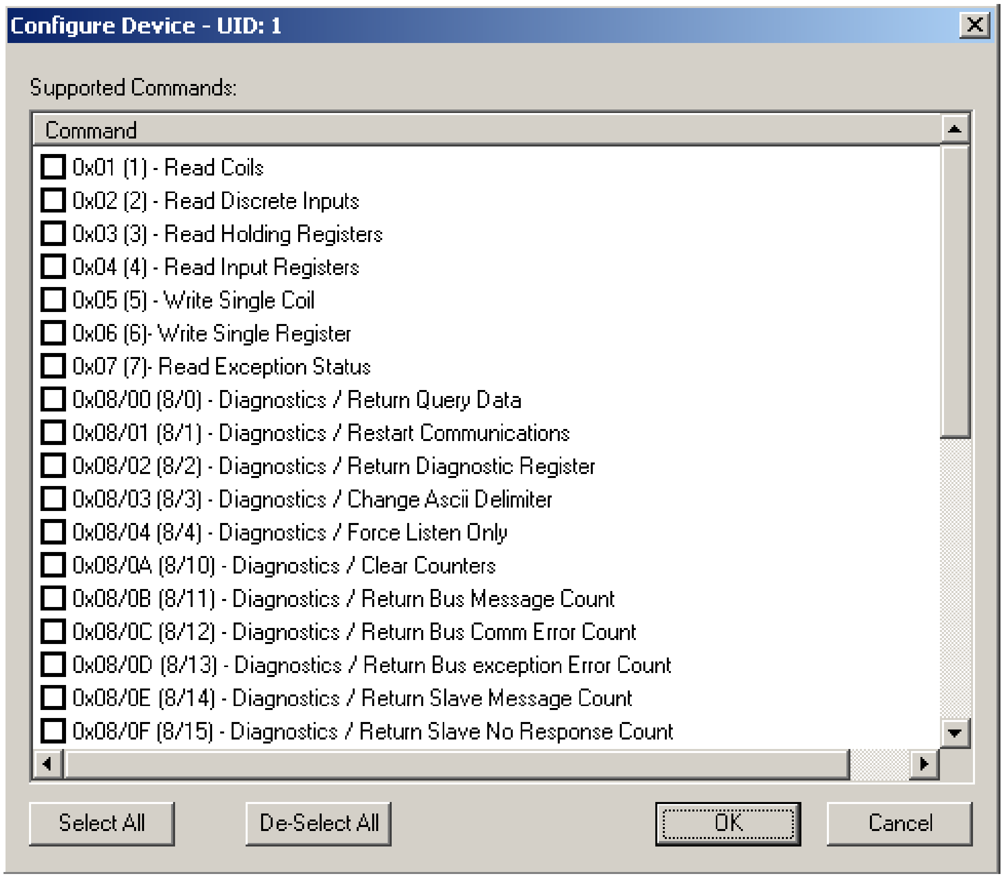


	Unit ID	Enabled	Timeout (ms)	Rate (ms)	Type	Test Mode	Test Addr	Data	Data Change
1	1	☑	0	0	Use Default	Use Default	0		Use Default
2	15	☑	0	0	Use Default	Use Default	0		Use Default
3	0	☑	0	0	Use Default	Use Default	0		Use Default
4	0	☑	0	0	Use Default	Use Default	0		Use Default
5	0	☑	0	0	Use Default	Use Default	0		Use Default
6	0	☑	0	0	Use Default	Use Default	0		Use Default
7	0	☑	0	0	Use Default	Use Default	0		Use Default
8	0	☑	0	0	Use Default	Use Default	0		Use Default
9	0	☑	0	0	Use Default	Use Default	0		Use Default
10	0	☑	0	0	Use Default	Use Default	0		Use Default
11	0	☑	0	0	Use Default	Use Default	0		Use Default
12	0	☑	0	0	Use Default	Use Default	0		Use Default
13	0	☑	0	0	Use Default	Use Default	0		Use Default
14	0	☑	0	0	Use Default	Use Default	0		Use Default
15	0	☑	0	0	Use Default	Use Default	0		Use Default
16	0	☑	0	0	Use Default	Use Default	0		Use Default

Tip

Do not select commands that the device does not support. This prevents unsupported commands from being issued and avoids lengthy timeouts from occurring that result in decreased performance and connection reconnects. Safety Manager Commands are automatically configured.

28 Select a given device in the grid and click the **Edit Commands** button to call up the **Configure Device** dialog and select the desired commands for the selected device.



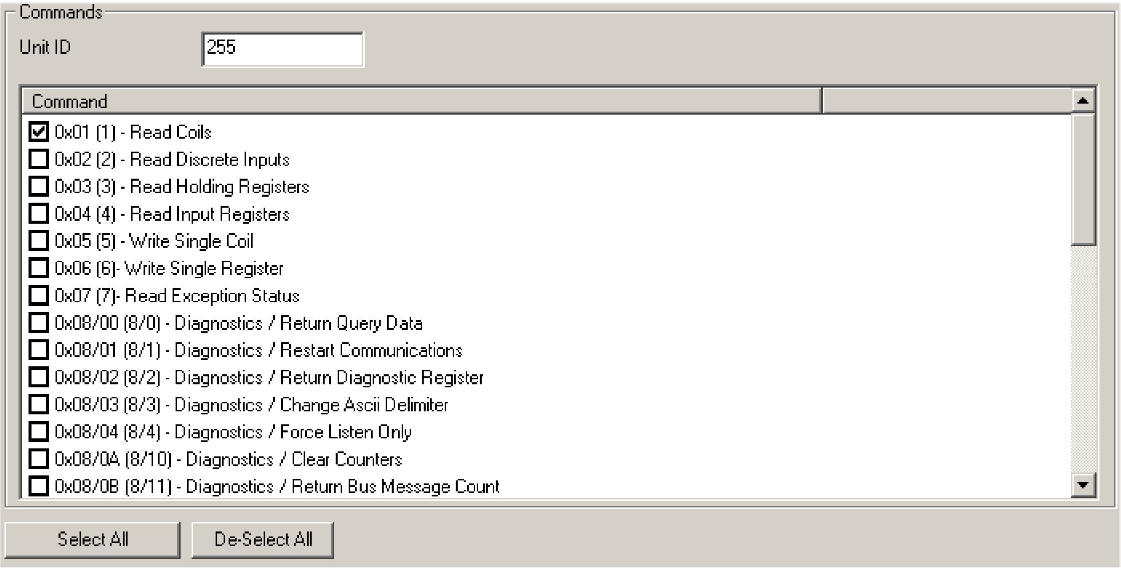
□ 0x01 (1) - Read Coils
□ 0x02 (2) - Read Discrete Inputs
□ 0x03 (3) - Read Holding Registers
□ 0x04 (4) - Read Input Registers
□ 0x05 (5) - Write Single Coil
□ 0x06 (6)- Write Single Register
□ 0x07 (7)- Read Exception Status
□ 0x08/00 (8/0) - Diagnostics / Return Query Data
□ 0x08/01 (8/1) - Diagnostics / Restart Communications
□ 0x08/02 (8/2) - Diagnostics / Return Diagnostic Register
□ 0x08/03 (8/3) - Diagnostics / Change Ascii Delimiter
□ 0x08/04 (8/4) - Diagnostics / Force Listen Only
□ 0x08/0A (8/10) - Diagnostics / Clear Counters
□ 0x08/0B (8/11) - Diagnostics / Return Bus Message Count
□ 0x08/0C (8/12) - Diagnostics / Return Bus Comm Error Count
□ 0x08/0D (8/13) - Diagnostics / Return Bus exception Error Count
□ 0x08/0E (8/14) - Diagnostics / Return Slave Message Count
□ 0x08/0F (8/15) - Diagnostics / Return Slave No Response Count

- Be sure to enter desired configuration data for each device connected to the bridge.
- Go to Step 29.

29 *Optional*

You can enter values in the grid fields to override the default behavior of the values configured in Steps 15 to 26.

30 The **Slave Configuration** tab for Honeywell's Safety Manager or a MODBUS TCP device type includes the **Unit ID** box and **Command** list box for configuring details about the device.



Commands

Unit ID: 255

Command

- ☑ 0x01 (1) - Read Coils
- □ 0x02 (2) - Read Discrete Inputs
- □ 0x03 (3) - Read Holding Registers
- □ 0x04 (4) - Read Input Registers
- □ 0x05 (5) - Write Single Coil
- □ 0x06 (6)- Write Single Register
- □ 0x07 (7)- Read Exception Status
- □ 0x08/00 (8/0) - Diagnostics / Return Query Data
- □ 0x08/01 (8/1) - Diagnostics / Restart Communications
- □ 0x08/02 (8/2) - Diagnostics / Return Diagnostic Register
- □ 0x08/03 (8/3) - Diagnostics / Change Ascii Delimiter
- □ 0x08/04 (8/4) - Diagnostics / Force Listen Only
- □ 0x08/0A (8/10) - Diagnostics / Clear Counters
- □ 0x08/0B (8/11) - Diagnostics / Return Bus Message Count

- With the cursor in **Unit ID** (**UNITID**) box, key in the device index number between 1 to 247 or 255 for the device the block represents. The default is 255 for native device or 1 for Safety Manager.
- Select the desired commands for the device in the Command list box. Use the Select ALL or De-Select All buttons, as applicable. See the next section “Device supported commands” on page 43 for more information.

31 Click the **Module Statistics** tab to display it.

- Check the **Alarming Enabled** (**ALMENBSTATE**) check box to have alarm conditions displayed in the Alarm Summary and Journals. Clear the check box to disable the display of alarm conditions. The default is checked.
- The other parameters on this form are read only, except for RSTSTATS button, in the **Monitoring** mode and are not configurable in **Project**. Error codes are displayed to assist users in tracing the cause of an error.

32 Click the **Connections Statistics** tab to display it.

The parameters on this form are read only, except for the RSTERRCNT button, in the **Monitoring** mode and are not configurable in **Project**.

33 Click the **Channel Status** tab to display it.

The parameters on this form are read only, except for the CHANRSTSTATS button, in the **Monitoring** mode and are not configurable in **Project**.

34 Click the **QVCS** tab to display it.

The parameters on this form are mostly read only and provide information about the Qualification and Version Control System pertinent to the device. See the *Control Building Guide* and/or the online help for more information.

35 Click the **Server History** tab to display it.

Use this tab to configure desired history parameters and create or edit server scripts. See the *Control Building Guide* and/or the online help for more information.

36 Click the **Server Displays** tab to display it.

Use this tab to configure parameters associated with Station displays. See the *Control Building Guide* and/or the online help for more information.

37 Click the Control Confirmation tab to display it.

Use this tab to configure parameters for control confirmation associated with the licensed Electronic Signature option. See the *Control Building Guide* and/or the online help for more information.

38 Click the **Identification** tab to display it.

Use this tab to configure parameters associated with the licensed template option. See the *Control Building Guide* and/or the online help for more information.

39 When you completed entering configuration data, click the **OK** button to close the configuration form and save the data.

40 This completes the procedure. Go to the next section.

5.3 Device supported commands

The following table lists the supported commands by function code along with a description and PCDI block usage as well as default settings for Safety Manager.

Table 1: Device Supported Commands

Function Code	Description	Used by PCDI Blocks	Safety Manager Default
0x01	Read Coils	PCDI_MASTER - Diag, PCDIFLAGARRCH	On
0x02	Read Discrete Inputs	PCDI_MASTER - Diag, PCDIFLAGARRCH	On
0x03	Read Holding Registers	PCDI_MASTER - Diag, PCDITEXTARRCH, PCDINUMARRCH	On
0x04	Read Input Registers	PCDI_MASTER - Diag, PCDITEXTARRCH, PCDINUMARRCH	On
0x05	Write Single Coil	PCDI_MASTER - Diag, PCDIFLAGARRCH	On
0x06	Write Single Register	PCDI_MASTER - Diag, PCDITEXTARRCH, PCDINUMARRCH	On
0x07	Read Exception Status	PCDI_MASTER - Diag	Off
0x08 / 0x00	Diagnostic - Return Query Data	PCDI_MASTER - Diag, PCDITEXTARRCH - loopback	On
0x08 / 0x01	Diagnostic - Restart Communications		Off
0x08 / 0x02	Diagnostic - Return Diagnostic Register	PCDI_MASTER - Diag	On
0x08 / 0x03	Diagnostic - Change ASCII Delimiter		Off
0x08 / 0x04	Diagnostic - Force Listen Only		Off
0x08 / 0x0A	Diagnostic - Clear Counters and Diagnostic Register		Off
0x08 / 0x0B	Diagnostic - Return Bus Message Count		Off
0x08 / 0x0C	Diagnostic - Return Bus Communication Error Count		Off
0x08 / 0x0D	Diagnostic - Return Bus Exception Error Count		Off
0x08 / 0x0E	Diagnostic - Return Slave Message Count		Off
0x08 / 0x0F	Return Slave No Response Count		Off
0x08 / 0x10	Return Slave NAK Count		Off
0x08 / 0x11	Return Slave Busy Count		Off
0x08 / 0x12	Return Bus Character Overrun Count		Off
0x08 / 0x13	Return IOP Message Overrun Count		Off
0x08 / 0x14	Get / Clear MODBUS Plus stats		Off
0x0B	Get Communication Event Counter		Off
0x0C	Get Communication Event Log	PCDI_MASTER - Diag	Off
0x0F	Write Multiple Coils	PCDIFLAGARRCH	On
0x10	Write Multiple Registers	PCDITEXTARRCH, PCDINUMARRCH	On

Function Code	Description	Used by PCDI Blocks	Safety Manager Default
0x11	Report Slave ID		Off
0x14 / 0x06	Read File Record		Off
0x15 / 0x06	Write File Record		Off
0x16	Mask Write Register		Off
0x17	Read / Write Multiple Registers		Off
0x18	Read FIFO Queue		Off
0x2B / 0x0E	Read Device Information	PCDI_MASTER - Honeywell Info	On

5.4 Assigning PCDI_Master Block to Execution Environment

You can also just drag and drop the PCDI_MASTER block to the desired CEEC300 folder in the **Project** tree instead of using the Execution Environment Assignment dialog, as outlined in the following procedure.

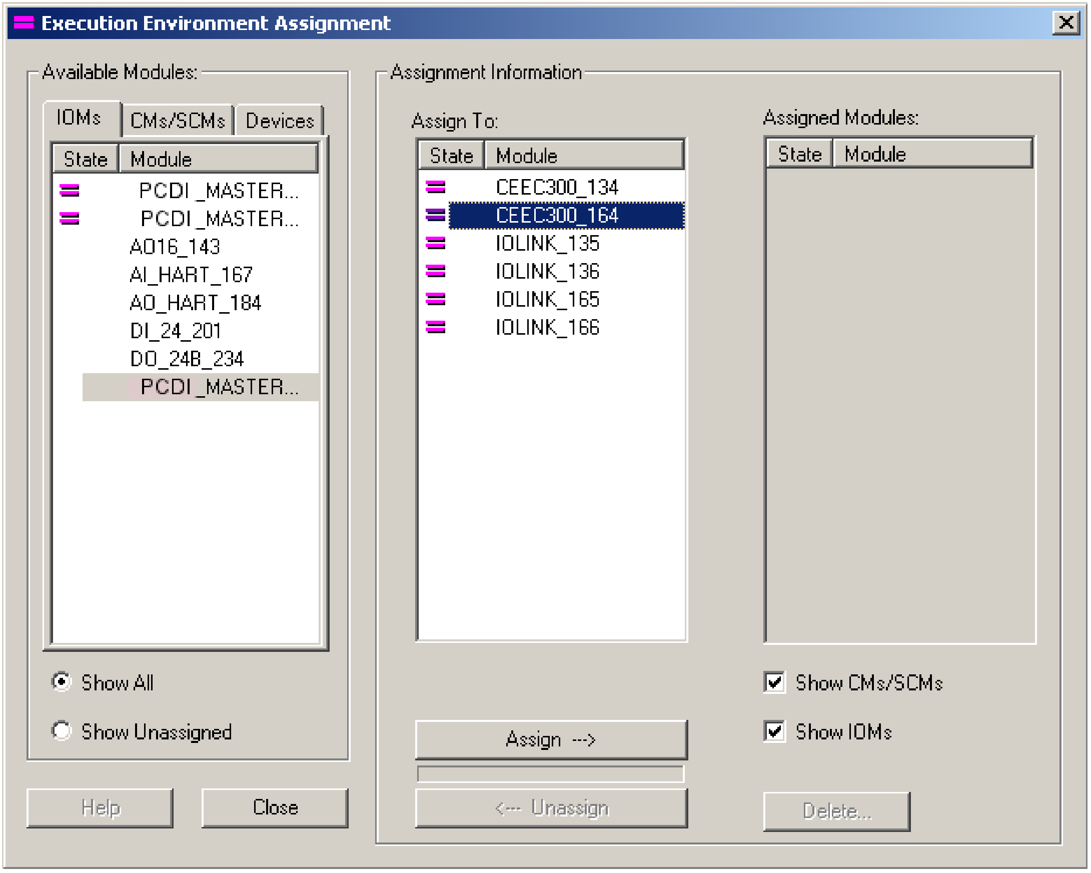
All illustrations used in the following procedure are for example purposes only.

Prerequisites

- You have started Configuration Studio and launched the Control Builder application.
- You have logged on with sufficient privileges to create control strategies using Control Builder.
- You have created a C300 Controller block in the **Project** tab.
- You have added a PCDI_MASTER block to the **Project** tab.

To assign PCDI_MASTER block to CEE

1 On the **Edit** menu, click **Execution Environment Assignment**; or click the Execution Environment Assignment button on the toolbar to call up the **Execution Environment Assignment** dialog, as shown in the sample illustration below.



Available Modules (IOMs):

State	Module
☰	PCDI _MASTER...
☰	PCDI _MASTER...
	AO16_143
	AI_HART_167
	AO_HART_184
	DI_24_201
	DO_24B_234
	PCDI _MASTER...

● Show All
○ Show Unassigned

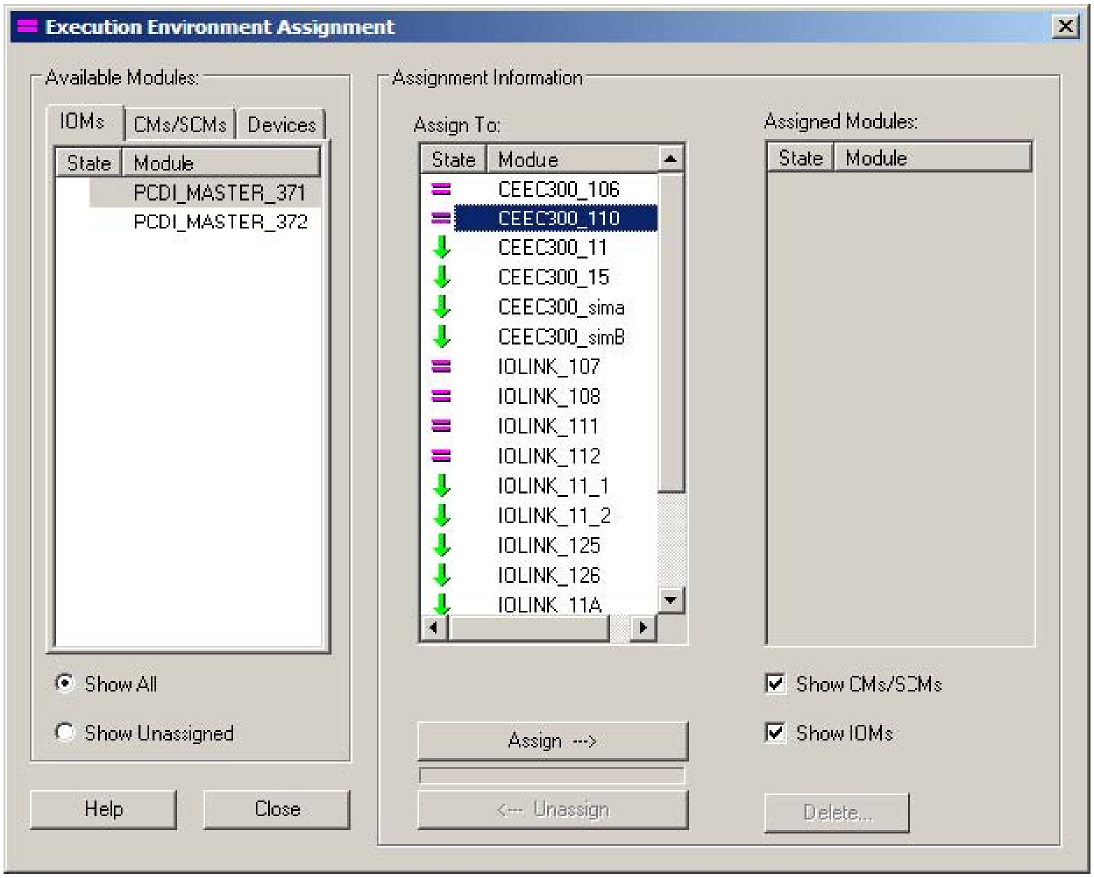
Assign To:

State	Module
☰	CEEC300_134
☰	CEEC300_164
☰	IOLINK_135
☰	IOLINK_136
☰	IOLINK_165
☰	IOLINK_166

Assigned Modules:

State	Module

☑ Show CMs/SCMs
☑ Show IOMs



Available Modules: IOMs

State	Module
	PCDI_MASTER_371
	PCDI_MASTER_372

● Show All

○ Show Unassigned

Assign To:

State	Modue
=	CEEC300_106
=	CEEC300_110
↓	CEEC300_11
↓	CEEC300_15
↓	CEEC300_sima
↓	CEEC300_simB
=	IOLINK_107
=	IOLINK_108
=	IOLINK_111
=	IOLINK_112
↓	IOLINK_11_1
↓	IOLINK_11_2
↓	IOLINK_125
↓	IOLINK_126
↓	IOLINK_11A

Assigned Modules:

State	Module

☑ Show CMs/SCMs

☑ Show IOMs

2 If the **IOMs** tab is not on display in the **Available modules** list box, click the tab to display it.
Click the PCDI_MASTER block to be assigned to select it.

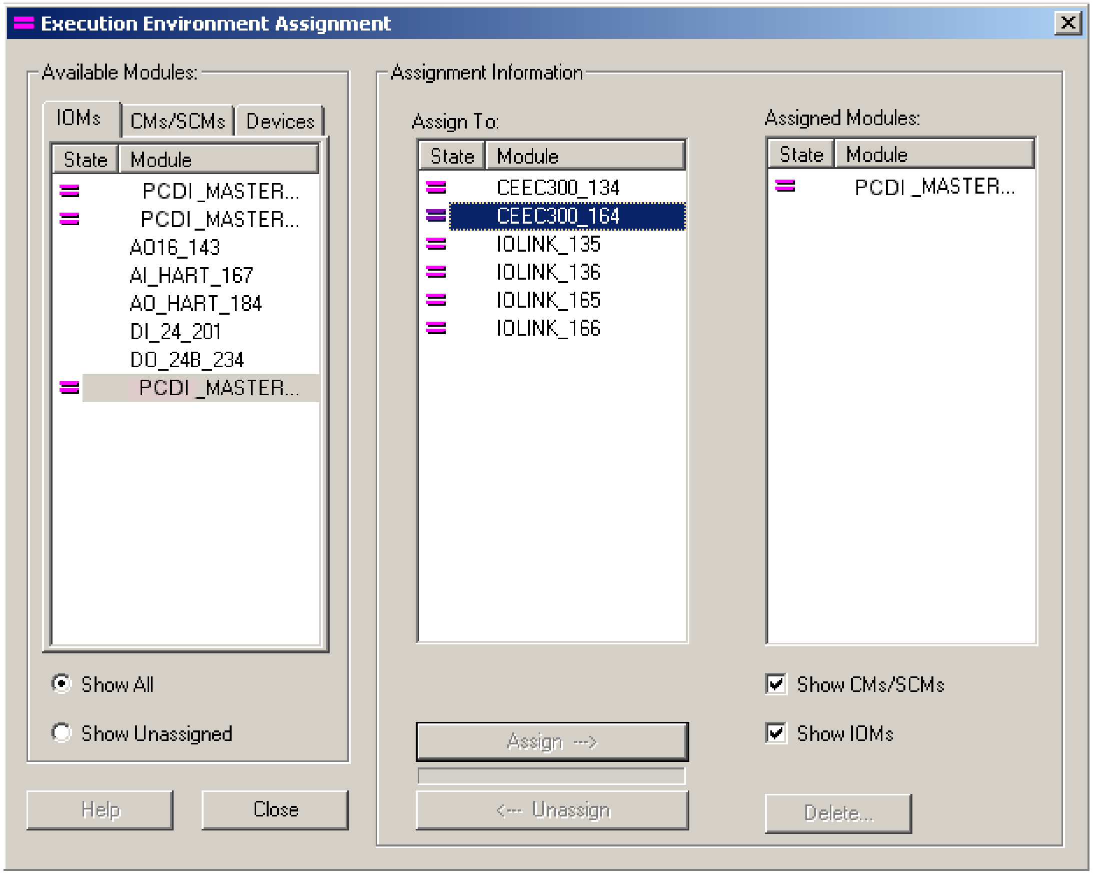
3 In the **Assign To** list box, click the CEE block that the PCDI_MASTER block is to be assigned to select it.

4 Click the **Assign ->** button to assign the selected module to the given CEE block.

5 Check that the PCDI_MASTER block now appears in the **Assigned Modules** list box for the given CEE block, as shown in the sample illustration below.
Note that the equal symbol

=

prefix means that the block has been assigned and the down arrow symbol ↓ prefix means that the block has been loaded to the Controller.



Available Modules (IOMs):

State	Module
▬	PCDI_MASTER...
▬	PCDI_MASTER...
	AO16_143
	AI_HART_167
	AO_HART_184
	DI_24_201
	DO_24B_234
▬	PCDI _MASTER...

● Show All

○ Show Unassigned

Assign To:

State	Module
▬	CEEC300_134
▬	CEEC300_164
▬	IOLINK_135
▬	IOLINK_136
▬	IOLINK_165
▬	IOLINK_166

Assigned Modules:

State	Module
▬	PCDI _MASTER...

☑ Show CMs/SCMs

☑ Show IOMs

6 Click the **Close** button to exit the dialog and save the assignments.

7 This completes the procedure. Go to the next section.

5.5 Adding PCDI Array Request Channel Block to Control Module

The following procedure adds a PCDI Numeric Array Request Channel (PCDINUMARRCH) block to a CM and assigns a channel to a PCDI_MASTER block for example purposes only. You can easily adapt this procedure to apply to a PCDI Flag Array Request Channel (PCDIFLAGARRCH) block or a PCDI Text Array Request Channel (PCDITEXTARRCH) block.

Each PCDI Array Request Channel block is automatically assigned a default tag name when it is created, which must only be unique within the containing Control Module. If your system will include multiple Array Channel blocks, you may want to adopt a more structured syntax for naming them.

The request block channel name can be up to 15 characters long and must contain at least on letter (A-Z). It must not contain an embedded space or leading space, and dots are allowed in parameter naming only.

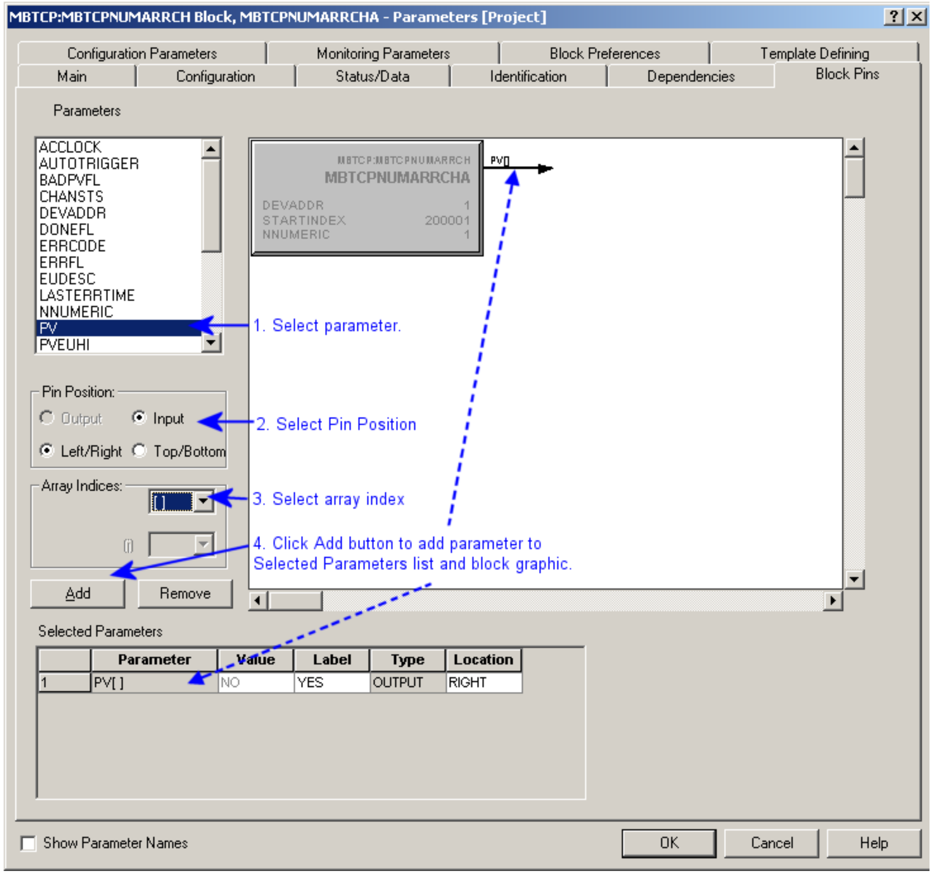
PCDI Numeric Array Channel (PCDINUMARRCH) function block's floating-point values may display rounding errors if the decimal (non-real) numbers are used for scaling/substitution. For example, if a PVEUHI value is 9.9 and PV is 4.0, then after scaling the PV value changes to 3.99. Similarly, if the PVEUHI is 10 and PV is 3.99, then after scaling the PV value changes to 4.0.

You should use either the *WriteAlwaysWriteOnly* or *WriteOnChgWriteOnly* Write Option (WRITEOPT) selection when configuring communications for a Safety Manager device. This avoids the possibility of wrong values being sent back as a result of a read carried out directly after a write.

If the Safety Manager is configured to stop/freeze if a communication error occurs, then write requests are not allowed to the Safety Manager until it has been reset. In addition, writes to the Safety Manager are not allowed, even after the Safety Manager is reset, if the WRITEOPT is configured as WriteOnChg and the Safety Manager device is running with faults. In this scenario, there is no indication of the writes to the Safety Manager failing because the write requests are in a queue. Because of this, the PCDI reports a 'Safety Manager Running With Faults diagnostic' alarm. A write to Safety Manager occurs when the WRITEOPT changes to WriteOnChg in the Monitoring view. Writes to Safety Manager can also occur when the WRITEOPT is set to one of the following write options.

- WriteAlwaysWriteOnly
- WriteOnChgWriteOnly
- WriteOnDiff
- WriteAlways

The Array Request Channel blocks are designed to read or write array data. Array data are accessible on a single exposed pin as **whole array** data, or on individual pins specified by array index, which is configurable through the **Block Pins** tab on the Array Request Channel block configuration form. A sample block pin configuration is shown in the following illustration.



MBTCP:MBTCPNUMARRCH Block, MBTCPNUMARRCHA - Parameters [Project]

Pin Position: ○ Output ● Input

● Left/Right ○ Top/Bottom

Array Indices: [ ]

1. Select parameter.

2. Select Pin Position

3. Select array index

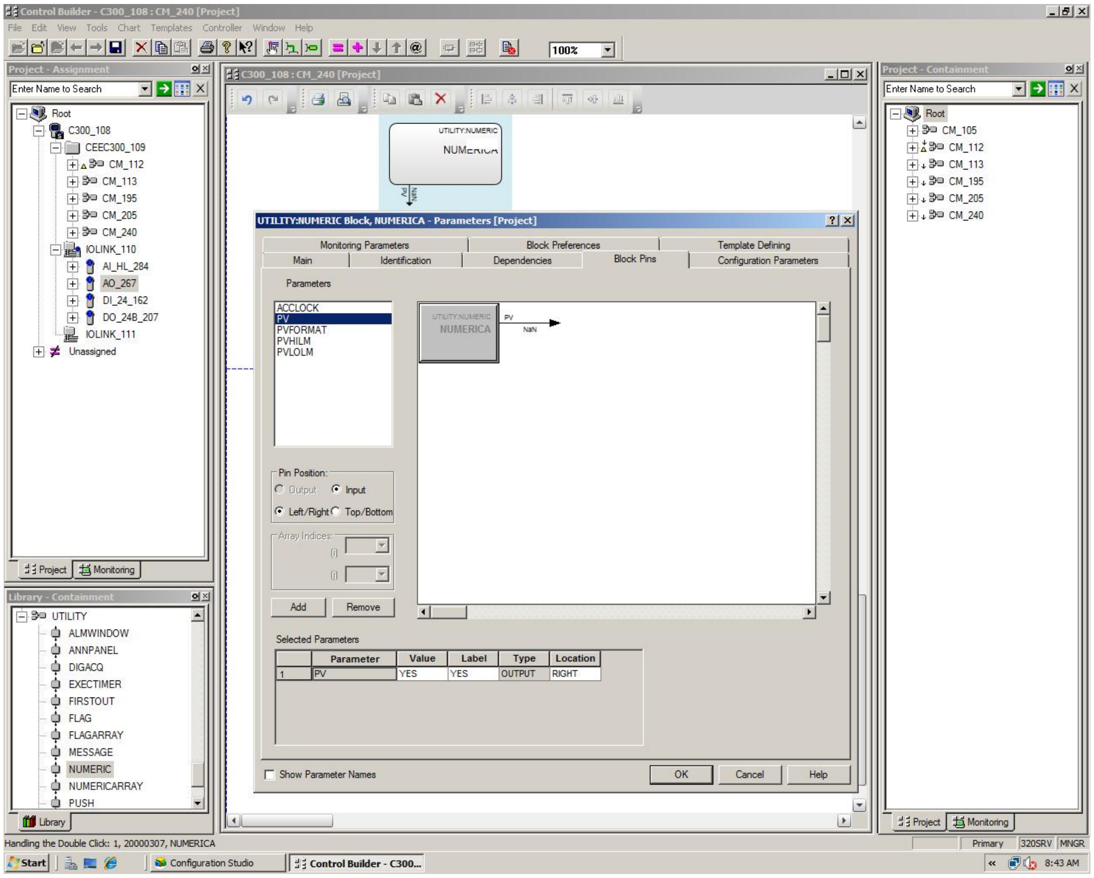
4. Click Add button to add parameter to Selected Parameters list and block graphic.

Selected Parameters

	Parameter	Value	Label	Type	Location
1	PV[ ]	NO	YES	OUTPUT	RIGHT

□ Show Parameter Names

Figure 4: Sample Block Pin Configuration for Whole Array Access.



Pin Position: ○ Output ● Input

● Left/Right ○ Top/Bottom

Selected Parameters

	Parameter	Value	Label	Type	Location
1	PV	YES	YES	OUTPUT	RIGHT

□ Show Parameter Names

Handling the Double Click: 1, 20000307, NUMERICA

Users are responsible for verifying that the correct MODBUS addresses are maintained between request and device blocks. There is no design time validation to verify the following:

- That the PCDI_MASTER Block device identifiers are synchronized with the Array Request Channel block device identifiers.
- That valid addresses are provided for devices.

For PCDI_MASTER block to appear in the drop-down list on the Array Channel block's configuration form, the Control Module containing the Array Channel block must be assigned to the same CEE as the PCDI_MASTER block or both be *unassigned* in Project.

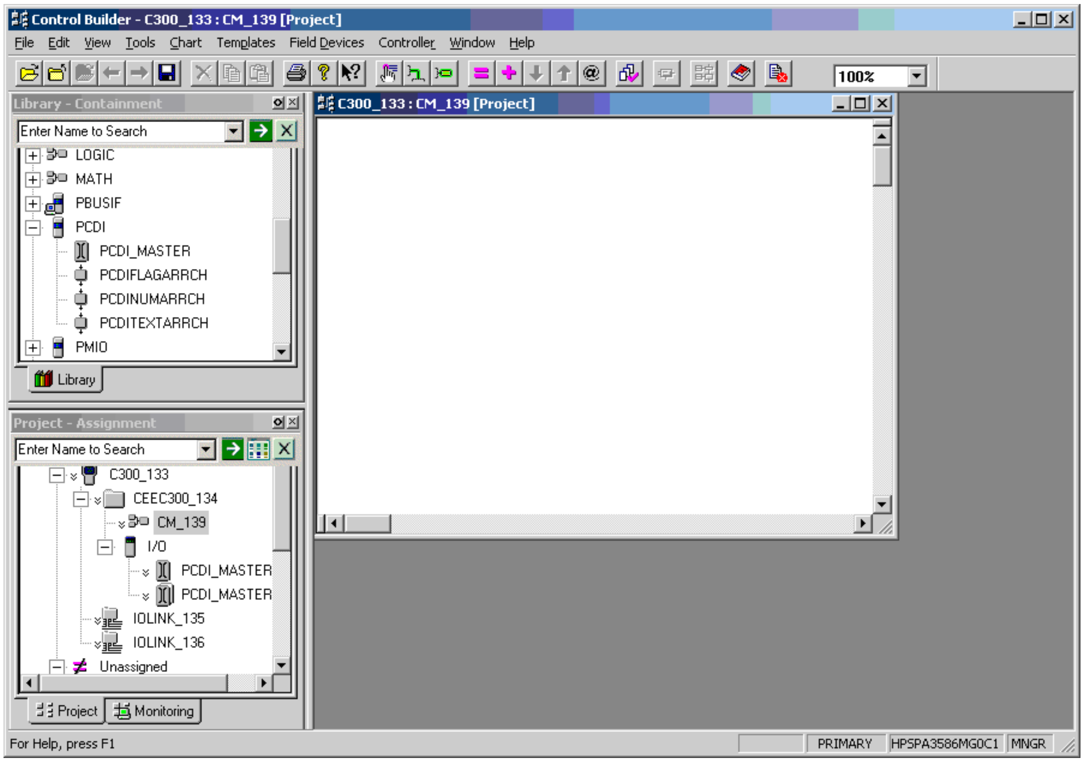
All illustrations used in the following procedure are for example purposes only.

Prerequisites

- You have started Configuration Studio and launched the Control Builder application.
- You have logged on with sufficient privileges to create control strategies using Control Builder.
- You have added a PCDI_MASTER block to the **Project** tab and assigned it to a C300 Control Execution Environment.
- You have created a Control Module in the **Project** tab and assigned it to the same C300 Control Execution Environment as the PCDI_MASTER block.

To add MODBUS TCP array request channel block to CM

1 In the **Project** tab, double-click the icon for the Control Module that is to include an array request channel block to open it in the Control Drawing pane, as shown in the following sample illustration.



2 In the **Library** tab, scroll to the PCDI icon and click the plus sign to expand the tree.

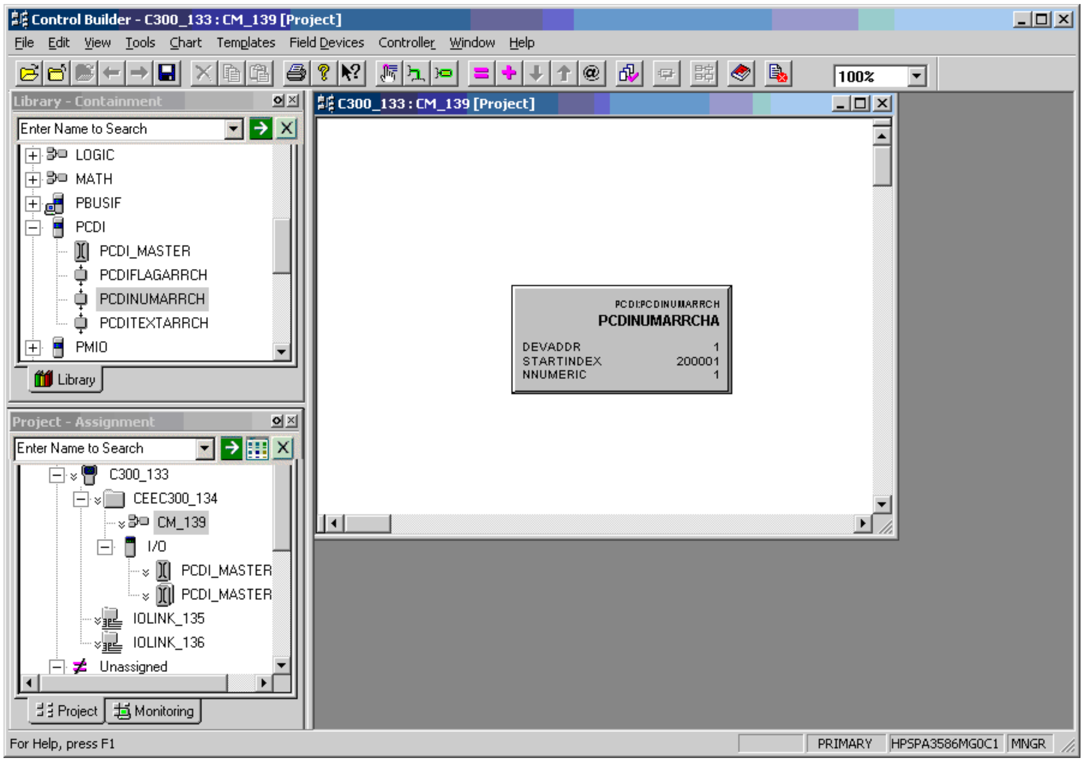
You can also select the tag name in the following ways.

- Select Library mode and type the initial characters of a tag name.
- Using the Search Option toolbar
- Type the initial few characters of a tag name in the Search Option toolbar. A list of all matching tag names appears.
- Select the tag name and click ➔ .

If the tag exists, the tree expands and the specified tag name is highlighted.

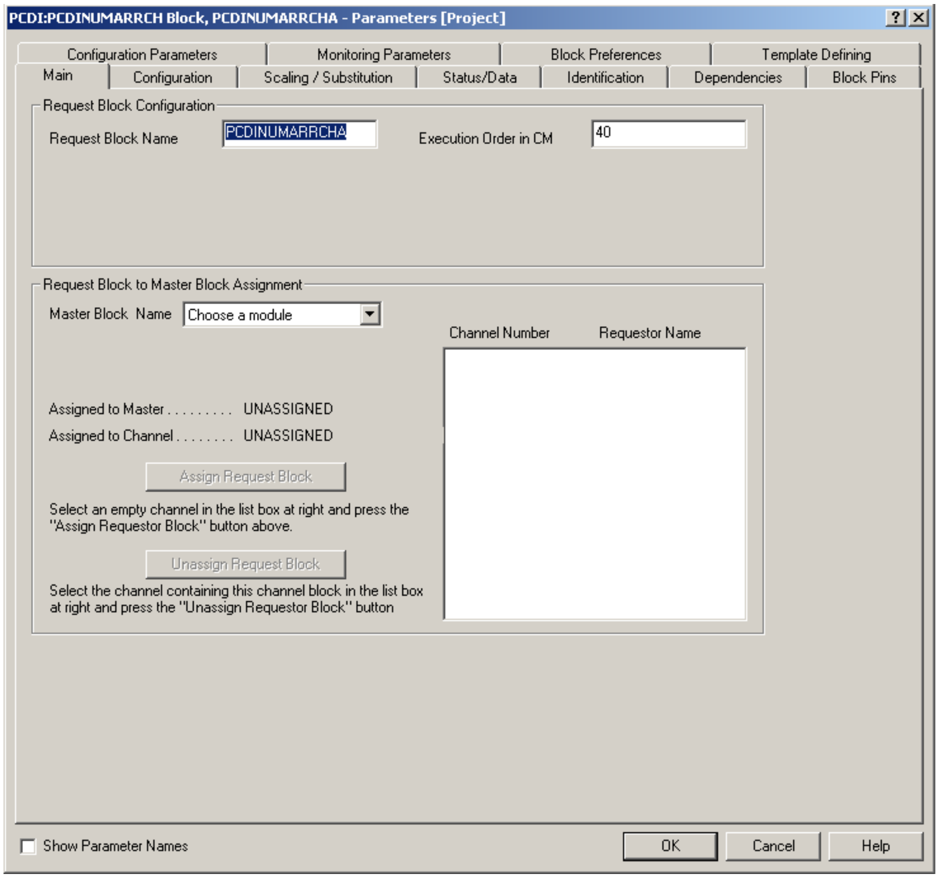
For more information on searching the tags, see *Control Building User's Guide.*

3 Click and drag the icon for the PCDINUMARRCH block to the open CM in the Control Drawing to add the block to the CM, as shown in the following sample illustration.



DEVADDR 1
STARTINDEX 200001
NNUMERIC 1

4 Double-click the PCDINUMARRCHA block to open its configuration parameters form.



Request Block Name PCDINUMARRCHA

Execution Order in CM 40

Master Block Name Choose a module

Assigned to Master . . . . UNASSIGNED

Assigned to Channel . . . . UNASSIGNED

Select an empty channel in the list box at right and press the "Assign Requestor Block" button above.

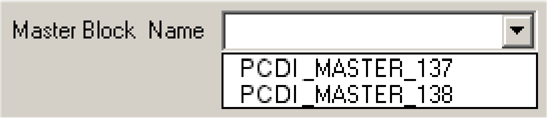
Select the channel containing this channel block in the list box at right and press the "Unassign Requestor Block" button

□ Show Parameter Names

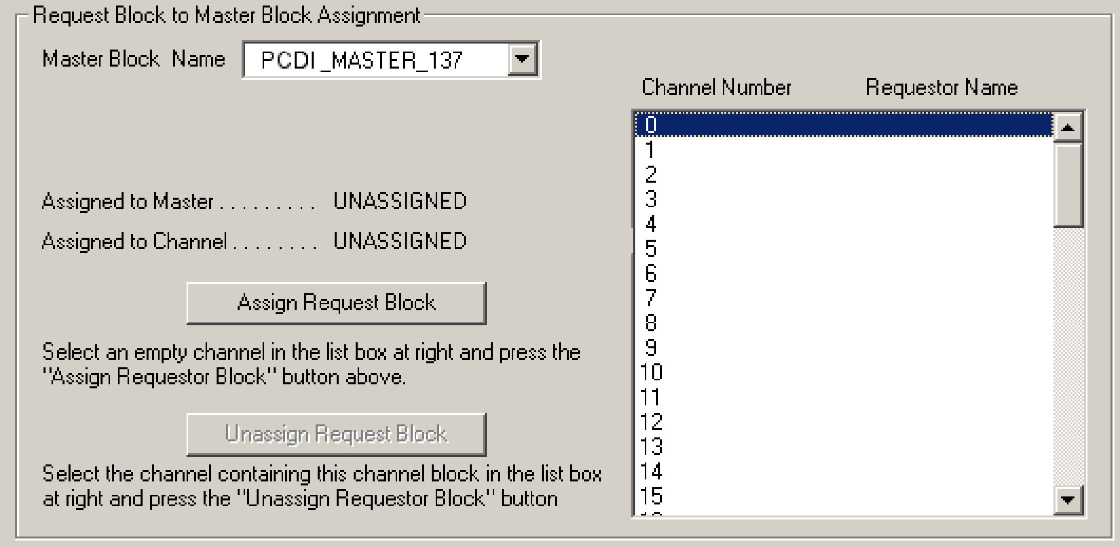
5 Accept the highlighted default **Channel Name** or key in a new one.

6 Double-click the **Execution Order in CM** box and key in the desired number in multiples of 10 to define the execution order of the block in the CM.

7 Click the down arrow in the **Master Block Name** box and select the PCDI_MASTER block to be associated with this channel.



Once the PCDI_MASTER is selected, the **Channel Number** field is automatically updated, as shown in the following sample illustration.



Master Block Name: PCDI_MASTER_137

Assigned to Master . . . . UNASSIGNED

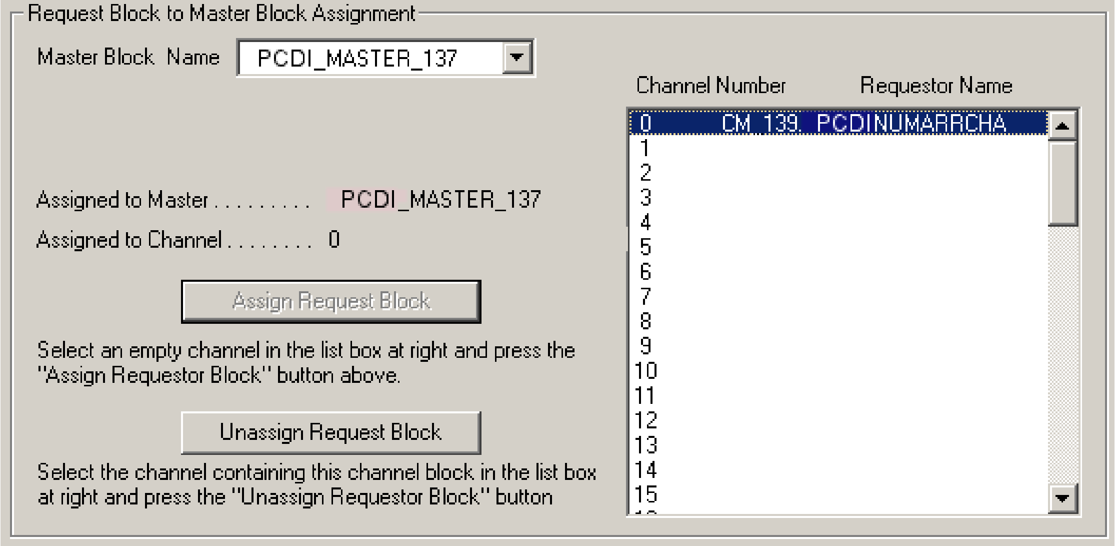
Assigned to Channel . . . . UNASSIGNED

Select an empty channel in the list box at right and press the "Assign Requestor Block" button above.

Select the channel containing this channel block in the list box at right and press the "Unassign Requestor Block" button

8 In the **Channel Number** list box, click the desired channel number this array request channel block is to be assigned to. Channel **0** or the lowest available channel number is the default selection.

9 Click the **Assign Request Block** button to assign this channel to the selected channel number. The **Requestor Name** field, the **Assigned to Master** and **Assigned to Channel** fields are automatically updated, as shown in the following sample illustration.



Master Block Name: PCDI_MASTER_137

0 CM_139. PCDINUMARRCHA

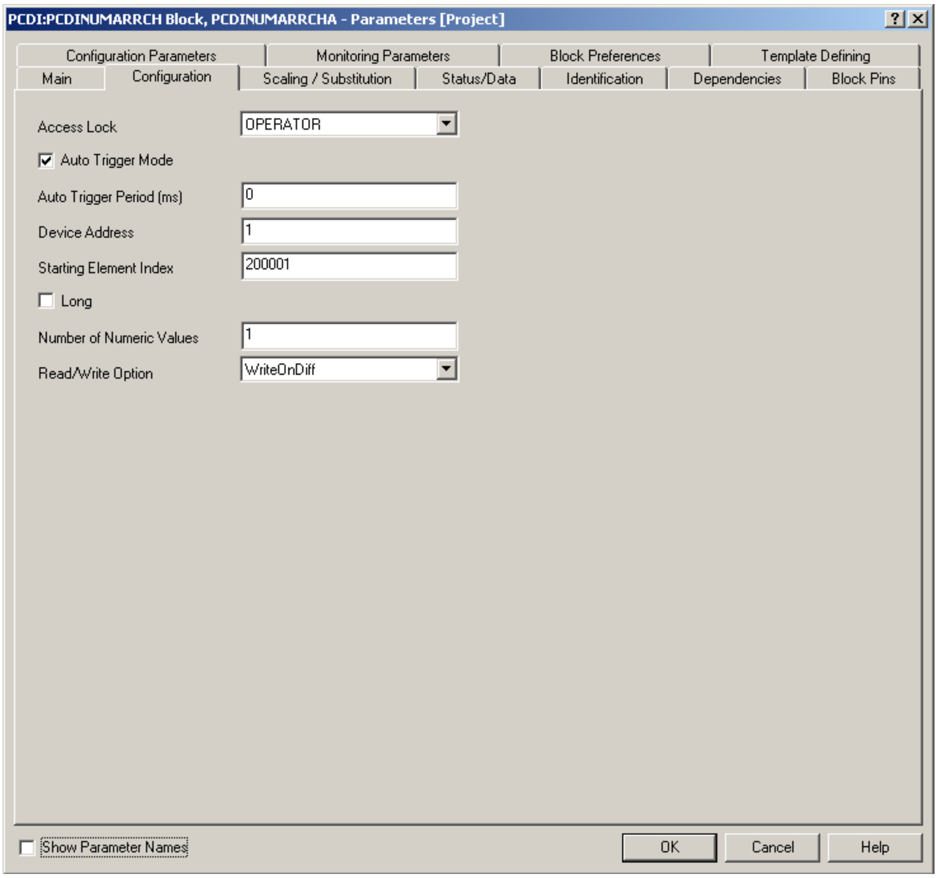
Assigned to Master . . . . PCDI_MASTER_137

Assigned to Channel 0

Select an empty channel in the list box at right and press the "Assign Requestor Block" button above.

Select the channel containing this channel block in the list box at right and press the "Unassign Requestor Block" button

10 Click the **Configuration** tab to display it.



PCDI:PCDINUMARRCH Block, PCDINUMARRCHA - Parameters [Project]

Configuration Parameters | Monitoring Parameters | Block Preferences | Template Defining

Main | Configuration | Scaling / Substitution | Status/Data | Identification | Dependencies | Block Pins

Access Lock: OPERATOR

☑ Auto Trigger Mode

Auto Trigger Period (ms): 0

Device Address: 1

Starting Element Index: 200001

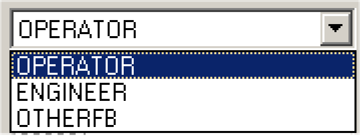
□ Long

Number of Numeric Values: 1

Read/Write Option: WriteOnDiff

□ Show Parameter Names

11 Click the down arrow in the **Access Lock (ACCLOCK)** box and select who can change the block's value or state based on sign on privileges. The default is Operator. Press the **Tab** key to move the cursor to the **Module Device Address** box.



12 Check the **Auto Trigger Mode (AUTOTRIGGER)** check box to enable it. Clear the check box to disable it. The default is checked so the block operates in the auto-triggered mode rather than the triggered mode. Press the **Tab** key to move the cursor to the **Auto Trigger Period (ms) (AUTOTRIGGERPERIOD)** box

13 Key in the time in milliseconds to delay the beginning of the following auto-triggered block execution. The default value is 0 milliseconds, which results in the block being triggered as quickly as possible. Press the **Tab** key to move the cursor to the **Device Address (DEVADDR)** box.

(*Auto triggered execution will resume in the Control Module cycle following expiration of the auto trigger period. If the auto trigger period has expired before the DONEFL parameter transitions to true, the block will be auto triggered immediately*.)

14 Key in the address that identifies the MODBUS device connected to this request channel. The default is 1. Press the **Tab** key to move the cursor to the **Starting Element Index** (**STARTINDEX**) box.

15 Key in the value to identify the start of the contiguous array in the MODBUS device. See the following *Starting Element Index Values* section for more information about configuring the start index value. The default varies for the given request channel type:

- PCDIFLAGARRCH = 1
- PCDINUMARRCH = 20001
- PCDITEXTARRCH = 500001

16 For PCDINUMARRCH block only:

Check the **LONG (CONVTOLONG)** check box to represent numeric values as longs. The default is unchecked so numeric values are represented as floating point values.

(*Checking the Long check box, provides support for 32-bit signed or unsigned integers depending on the start index specified. The Long check box has no impact on floating point numerics, which are always 32-bit. See the Startindex range 900001 - 965535 in Start index ranges define MODBUS function table for more information.)*

17 For PCDIFLAGARRCH block only:

With cursor in **Number of Flag Values (NFLAG)** box, key in the number of Boolean values in the array of data from the device. The default is 1.

18 For PCDITEXTARRCH block only:

Check the **ASCII Conversion (CONVTOASCII)** box to use ASCII characters. Clear the check box to use Unicode characters. The default is checked to use ASCII characters.

19 For PCDINUMARRCH block only:

With cursor in the **Number of Numeric Values (NNUMERIC)** box, key in the number that represents the number of numeric values in the array of data from the device. The default is 1.

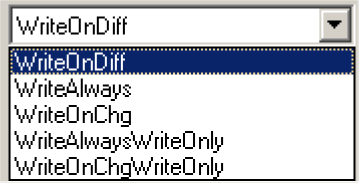
20 For PCDITEXTARRCH block only:

With cursor in **Number of String Values (NSTRING)** box, key in the number that represents the number of string values in the array of data from the device. The default is 1.

21 For PCDITEXTARRCH block only:

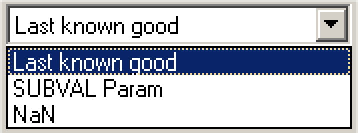
With cursor in **Char Length of String Values (STRLEN)** box, key in the number of characters per string value. The default is 8.

22 Click the down arrow in the **Write Option** (**WRITEOPT**) box and select how the Array Request Channel block will handle writes to its 'data' parameter. The default is WriteOnDiff.



23 Click the **Scaling/Substitution** tab to display the dialog. (This tab is titled just **Substitution** for PCDIFLAGARRCH and PCDITEXTARRCH blocks.)

24 Click the down arrow in the **Default Substitution Type** (**SUBVALTYPE**) box and select the value to substitute when an error occurs. The default is Last known good.



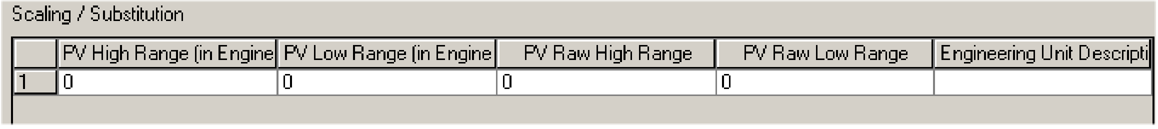
(*The NaN selection is only available for PCDINUMARRCH blocks*.)

25 For PCDINUMARRCH and PCDITEXTARRCH blocks only:

With cursor in **Default Substitution Value** (**SUBVAL**) box, key in desired value to substitute when an error occurs and the SUBVALTYPE setting is SUBVAL Param.

26 For PCDIFLAGARRCH block only: Check the **Set to ON on error** (**SUBVAL**) check box to turn the flag ON when an error occurs. Leave the check box blank to disable the function. The default is unchecked or disabled.

27 For PCDINUMARRCH block only: Key in applicable range information in the appropriate fields in the **Scaling/Substitution** grid.

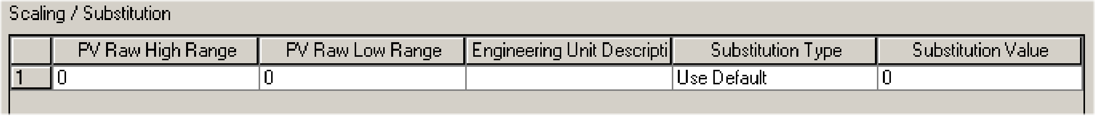


Scaling / Substitution

	PV High Range (in Engine	PV Low Range (in Engine	PV Raw High Range	PV Raw Low Range	Engineering Unit Descripti
1	0	0	0	0	

(*Leaving all values at zero will perform raw writes/reads of PV data. Input values will be limited by the range of the data type selected by the start index entered. Entering PVEUHI and PVEULO values only will limit the input values by data type and PVEUHI/PVEULO values. Entering both PVEUHI/LO and PVRAWHI/LO values will both limit the input and translate the input into the specified PVRAW range. If PVRAWHI/LO values are specified, the PVRAW value written will be limited to the specified range. PV output values are always reported as read with unlimited scaling applied. Note that no alarms are generated when values go over/under limits, since there is no alarming associated with the concept of MODBUS scaling.*)

a Click and move the scroll bar at the bottom of the grid to the right side to expose the **Substitution Type (ELEMSUBVALTYPE)** and **Substitution Value (ELEMSUBVAL)** fields.



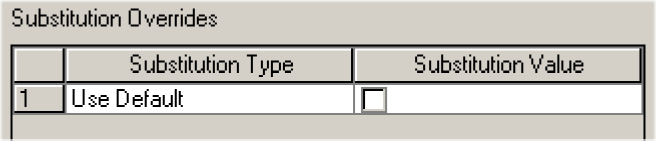
Scaling / Substitution

	PV Raw High Range	PV Raw Low Range	Engineering Unit Descripti	Substitution Type	Substitution Value
1	0	0		Use Default	0

b Click in the desired row of the **Substitution Type (ELEMSUBVALTYPE)** column and select the desired type from the list. This setting will override the SUBVALTYPE setting unless it is set to Use Default. The default is Use Default, which means the SUBVALTYPE setting is used.

c Click the desired row of the **Substitution Value (ELEMSUBVAL)** column and key in the desired value to be used when ELEMSUBVALTYPE is SUBVAL Param.

28 For PCDIFLAGARRCH block only: Key in applicable information in the appropriate fields in the **Substitution Overrides** grid.



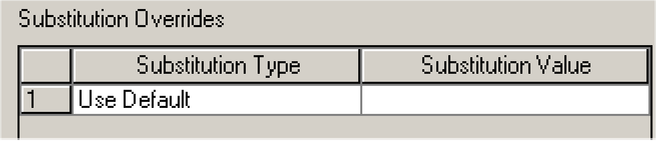
Substitution Overrides

	Substitution Type	Substitution Value
1	Use Default	☐

a Click in the desired row of the **Substitution Type (ELEMSUBVALTYPE)** column and select the desired type from the list. This setting will override the SUBVALTYPE setting unless it is set to Use Default. The default is Use Default, which means the SUBVALTYPE setting is used.

b Click the desired row of the **Substitution Value (ELEMSUBVAL)** column and check the check box to use ON or leave it unchecked for OFF as the value to use when ELEMSUBVALTYPE is SUBVAL Param.

29 For PCDITEXTARRCH block only:Key in applicable information in the appropriate fields in the **Substitution Overrides** grid.



Substitution Overrides

	Substitution Type	Substitution Value
1	Use Default	

a Click in the desired row of the **Substitution Type (ELEMSUBVALTYPE)** column and select the desired type from the list. This setting will override the SUBVALTYPE setting unless it is set to Use Default. The default is Use Default, which means the SUBVALTYPE setting is used.

b Click the desired row of the **Substitution Value (ELEMSUBVAL)** column and key in the desired value to be used when ELEMSUBVALTYPE is SUBVAL Param.

30 Click the **Status/Data** tab to display the dialog.

There are no parameters on this tab to configure in **Project**. Use this tab to monitor selected data in the **Monitoring** mode. The PCDI_MASTER block and Control Module must be loaded and active to make data available on this tab.

31 Click the **Identification** tab to display the dialog.

Use this tab to configure parameters associated with the licensed template option. See the *Control Building Guide* and/or the online help for more information.

32 Click the **Block Pins** tab to display the dialog.

Use this tab to configure the pins you want to expose for this block. See the *Control Building Guide* and/or the online help for more information.

33 Click the **Configuration Parameters** tab to display the dialog.

Use this tab to configure the parameters to be displayed on the block in the **Project** mode. See the *Control Building Guide* and/or the online help for more information.

34 Click the **Monitoring Parameters** tab to display the dialog.

Use this tab to configure the parameters to be displayed on the block in the **Monitoring** mode. See the *Control Building Guide* and/or the online help for more information.

35 Click the **Block Preferences** tab to display the dialog.

Use this tab to configure the viewing preferences for the block. See the *Control Building Guide* and/or the online help for more information.

36 If you have a template license, click the **Template Defining** tab to display the dialog.

Use this tab to select which parameters are to be template defining. See the *Control Building Guide* and/or the online help for more information.

37 When you completed entering configuration data, click the **OK** button to close the configuration form and save the data.

38 This completes the procedure. Go to the next section.

5.6 Starting Element Index Values

You specify the Starting Element Index (STARTINDEX) parameter through the **Configuration** tab on the configuration form for the given PCDI Array Request Channel block in Control Builder, as noted in the previous procedure. This parameter defines the MODBUS address map.

The value configured for the STARTINDEX parameter in a given Array Request Channel block determines the MODBUS Start Index and function. Only one type of external data can be accessed per Array Channel block.

The STARTINDEX parameter specifies the MODBUS data type and a MODBUS address up to 65535. The most significant digit identifies the MODBUS function. For example, read coil status as noted in the following table. The lower five digits define the starting address to access data for the given MODBUS function. For example, the address ranges 2xxxxx, 4xxxxx, 5xxxxx, 7xxxxx, 8xxxxx, and 9xxxxx all address the same xxxxx holding registers in the MODBUS end device, but with different data formats. When multiple numeric data types are used, you must configure the STARTINDEX for each array to avoid overlapping arrays.

The following table shows the relationship between MODBUS functions, the STARTINDEX parameter range, and number of elements. The maximum number of elements specifies the maximum number of flags, numerics, or strings that can be configured for a given STARTINDEX range and Array Request Channel block type. All of the specified elements are read in a single transaction.

Table 2: Start index ranges define MODBUS function

STARTINDEX Ranges	MODBUS Function	Read FC Write FC	Array Request Channel Block	Data Range	Max. No. Elem.
000001 - 065535	Read Coil Status	01	PCDIFLAGARRCH	ON/OFF	1968
	Write Single Coil	05	Read/Write Bit Flags		
	Write Multiple Coils	15			
100001 - 165535	Read Input Status	02	PCDIFLAGARRCH	ON/OFF	1968
	Not Applicable	N/A	Read Only Bit Flags		
200001 - 265535	Read Holding Registers	03	PCDINUMARRCH	-34E38 to 34E38	60
	Write Single Register	06	Word Swapped IEEE FI.Pt		
	Write Multiple Registers	16	Read/Write		
300001 - 365535	Read Input Registers	04	PCDINUMARRCH	-32768 to 32767	120
	Not Applicable	N/A	Signed Integer Words Read only	-2147483648 to 2147483647	60
400001 - 465535	Read Holding Registers	03	PCDINUMARRCH	-32768 to 32767	120
	Write Single Register	06	Signed Integer Word/Long Array	-2147483648 to 2147483647	60
	Write Multiple Registers	16	Read/Write		
500001 - 565535	Read Holding Registers	03	PCDITEXTARRCH	Printable Characters	See
	Write Single Register	06	ASCII Encoded Bytes or UNICODE (2 Byte) Characters		Note 1
	Write Multiple Registers	16	Read/Write		
600000	Loopback Diagnostics	08	PCDITEXTARRCH Loopback Test	Printable Characters	See Note 1

STARTINDEX Ranges	MODBUS Function	Read FC Write FC	Array Request Channel Block	Data Range	Max. No. Elem.
600001 -665535	Read/Write Honeywell Information	03/06/16	Honeywell Information Transfer to Master Block		80 chars.
699999	Write Honeywell Information	06/16			
700001 - 765535	Read Holding Registers	03	PCDINUMARRCH IEEE Floating Point Read/Write	-34E38 to 34E38	60
	Write Single Registers	06			
	Write Multiple Registers	16			
800002 - 865535	Read Holding Registers	03	PCDINUMARRCH IEEE FI. Pt Read/Write	-34E38 to 34E38	60
	Write Single Registers	06			
	Write Multiple Registers	16			
900001 - 965535	Read Holding Registers	03	PCDINUMARRCH Unsigned Integer Words / Longs Read/Write	0 to 65535 0 to 4294967295	120 60[2]
	Write Single Register	06			
	Write Multiple Register	16			

1. NSTRING = 120 / STRLEN, if CONVTOASCII is not checked; or NSTRING = 240 / STRLEN, if CONVTOASCII is checked.
2. 60, if CONVTOLONG is selected; 120 if not.

5.7 MODBUS loopback diagnostics and Text Array Request Channel block configuration

The Text Array Request Channel block supports up to 240 bytes. If the CONVERTTOASCII parameter is enabled, it can be configured as 240 ASCII characters. If not enabled, it can be configured as 120 Unicode (wide-2 byte) characters.

You can divide the 240 bytes into as many as 16 strings by configuring the number of string (NSTRING) and string length (STRLEN) parameter values accordingly. The data transferred is always the entire string. All strings are SPACE character padded. There is no NULL padding

The 500001 - 565535 range implements the standard write/read Text Array Request Channel block functionality. All strings read into the STR[1..NSTRING] values will be space padded to the STRLEN parameter value.

Address 600000 implements a diagnostic loopback function. In this mode, data written to the block follows the same rules as for the Text Array Request Channel block, but the MODBUS function code used is for diagnostic - return query data (08-00). Returned data in the loopback response is saved to the output STR process value. There is no read request sent for address 600000.

Since the MODBUS function can only handle a single word (2 bytes) at a time, the number of writes required to complete the cycle will be depend on whether CONVERTTOASCII is enabled or not, the NSTRING value, and the STRLEN value.

For Addresses in the 600001 - 665535 range, values will be transferred to both the STR process value and will be written to the associated PCDI_MASTER block VENDOR string. The VENDOR string is 80 characters and will truncate excessively long strings. If multiple strings are configured (NSTRING > 1), the strings read will be concatenated with a SPACE character separating them. Only printable characters will be transferred to the VENDOR string.

Address 699999 provides a means of transferring input data on the STR[1..NSTRING] input pins into the PCDI_MASTER block VENDOR string. The same rules apply for the transfer of data as for the previous address range.

5.8 Whole array support

A whole array transfer allows you to connect together two arrayed parameters in Control Builder with the same data type. Once the configuration is loaded and running, the connection is formed between the two parameters and the entire array is requested and transferred to the destination block as one parameter access request. The array is available as a named parameter for connection to peer controllers by connecting to the parameter name.

Array write scenario

With whole array transfer, the write data can be brought in as an array from another block that supports whole array transfer. A whole array write can occur, if the write method is set to *WriteAlways*. With the other methods, the data write will be one element at a time.

The *WriteAlways* function will write all values transferred into the block. The *WriteOnDiff* and *WriteOnChg* functions will optimize use of single and multiple write commands to minimize the number of Control Module (CM) cycles required to transfer only those values that are different/changed. Use of single or multiple write commands is restricted by the commands selected in the Supported Commands dialog through the PCDI_MASTER block configuration form.

5.9 Simulation support

When you assign a PCDI_MASTER block and CM containing associated PCDI Array Request Channel blocks to the CEESIMC300FB, the blocks support the SIM-C300 simulation environment such that their input values can be read and written to by an operator or other applications. You can run the simulation of Array Request Channel blocks with or without a PCDI_MASTER block. For simulations without a PCDI_MASTER block, the alarms are not supported.

In the simulation environment, the communication between PCDI_MASTER block and MODBUS TCP hardware is blocked. You generate alarms manually by setting connection status, ERRFL, device status, and so on. The channel state for PCDI Array Request Channel blocks is put in *simulate* and is displayed on the channel block faceplate in Station and on the **Channel Status** tab of PCDI_MASTER block configuration form in the **Monitoring** tab of Control Builder.

You can configure the simulation mode as **DirectSub**, **SimValueSub**, or **None**. When mode is **DirectSub**, an operator or other application can write directly to the PV, ignoring the connection. When mode is **SimValueSub**, the SIMVALUE is substituted for the PV. The SIMVALUE can be fetched from another block using a parameter connection or it can be set by an operator or the other applications. In both modes, Error Flag and Error Code can be written by an operator or the other applications. When mode is **None**, the PV is fetched from connected blocks, or from applications or users with proper access level.

5.10 Loading configuration data to the CEE

The following procedure is based on doing an initial load of a block from the **Project** tab. You can also initiate a re-load of a block from the **Monitoring** tab.

You can load the CEE and its assigned contents such as I/O modules and Control Modules at the same time. See the *Control Building User's Guide* for more information about the load operation.

The PCDI Array Request Channel blocks are loaded with their containing Control Module.

All the illustrations used in the procedure are for example purposes only.

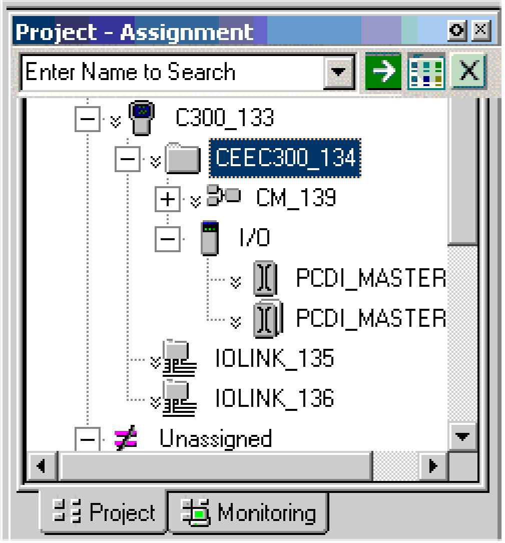
CAUTION	The load operation is still an offline function. The Load Dialog box provides the ability to automatically inactivate a component during a load and then return the component to its active state. Do **not** use this automatic inactivate/activate function, if your process can**not** tolerate the load disruption and consequent delay in activation. In this case, you must manually toggle the component state through the Monitoring tab in Control Builder.

Prerequisites

- You have started Configuration Studio and launched the Control Builder application.
- You have logged on with sufficient privileges to create control strategies using Control Builder.
- You have loaded the Controller and its execution environment.
- You have assigned the PCDI_MASTER block to the loaded Execution Environment

To load PCDI_MASTER block

1 In the **Project** tab, click the plus signs to expand the tree hierarchy for the C300 Controller, CEEC300, and I/O icons, as required.



You can also select the tag name in the following ways.

- Select Library mode and type the initial characters of a tag name.
- Using the Search Option toolbar

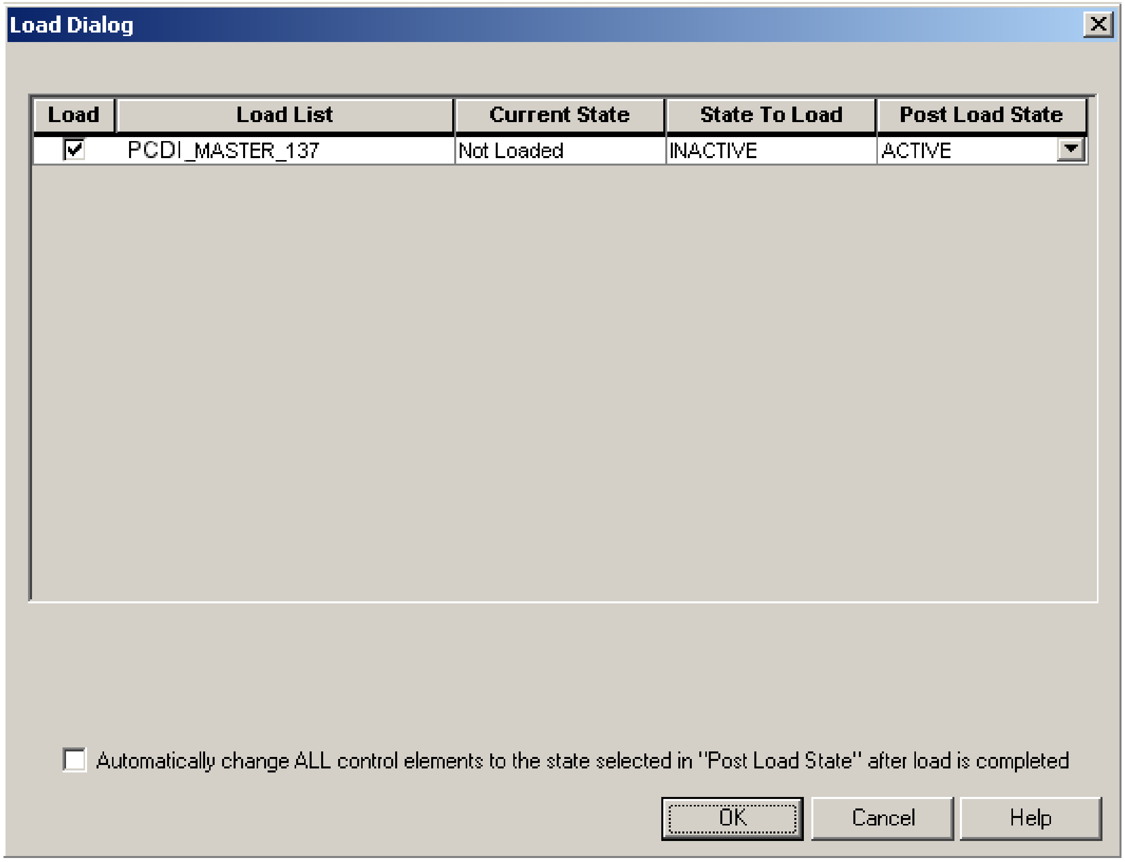
a Type the initial few characters of a tag name in the Search Option toolbar. A list of all matching tag names appears.

Select the tag name and click ➔

If the tag exists, the tree expands and the specified tag name is highlighted.

For more information on searching the tags, see *Control Building User's Guide*.

2 Right-click the PCDI_MASTER block icon and select **Load** from the menu to call up the Load dialog.



Load	Load List	Current State	State To Load	Post Load State
☑	PCDI_MASTER_137	Not Loaded	INACTIVE	ACTIVE

□ Automatically change ALL control elements to the state selected in "Post Load State" after load is completed

- Be sure the check box in the **Load** column is checked.
- Click the down arrow in the **Post Load State** column to select another state.
- Check the check box at the bottom of the dialog to automatically change the state of the PCDI_MASTER to the selected Post Load State after the load is completed.

3 Click the **OK** button to initiate the load and track progress through the **Load Monitor** dialog.

4 After load completes, check that the PCDI_MASTER icon now appears in the **Monitoring** tab tree hierarchy.

5 This completes the procedure. Go to the next section.

5.11 PCDI_MASTER Block properties form reference

For more details about a given parameter listed in one of the following sections, please refer to the Control Builder Parameter Reference.

Related topics

"Main tab parameters" on page 65
"Module Configuration tab parameters" on page 66
"Slave Configuration tab parameters" on page 67
"Module Statistics tab parameters" on page 68
"Connection Statistics tab parameters" on page 69
"Channel Status tab parameters" on page 70
"QVCS tab parameters" on page 71
"Server History tab parameters" on page 71
"Server Displays tab parameters" on page 72
"Control Confirmation tab parameters" on page 73
"Identification tab parameters" on page 73

5.11.1 Main tab parameters

The following table summarizes the parameter data you can monitor and/or configure on the Main tab of the configuration form for the selected PCDI_MASTER block. Remember to check the Show Parameter Names check box, if you want to display the parameter names instead of the plain text on the form.

Plain Text	Parameter Name	User Configurable	Notes
Module Name	NAME	Project Only	System assigned or user configured unique name. Consisting of up to 16 characters and at least one character must be a letter (A-Z).
Item Name	Item Name	Project Only	A non-unique name by which an entity is known within the context of the enterprise model.
Device Type	DEVTYPE	Project Only	Select the type of end device the block represents.

Plain Text	Parameter Name	User Configurable	Notes
Module Description	DESC	Project Only	User defined text that appears on Detail and Group displays for the block.
Currently Assigned Channels	N/A	Yes (through array request blocks)	Read only list box shows all Array Request Channel assignments to the block

5.11.2 Module Configuration tab parameters

The following table summarizes the parameter data you can monitor and/or configure on the **Module Configuration** tab of the configuration form for the selected PCDI_MASTER block. Remember to check the **Show Parameter Names** check box, if you want to display the parameter names instead of the plain text on the form.

Plain Text	Parameter Name	User Configurable	Notes
Primary IP Address	PRIMIP	**Project** Only	IP address for Primary or non-redundant Modbus device that block represents
Primary TCP Port	PRIMTCP	**Project** Only	All MODBUS application data units are sent through TCP to registered port 502.
Secondary IP Address	SECIP	**Project** Only	IP address for Secondary Modbus device. Only available for redundant Device Types.
Secondary TCP Port	SECTCP	**Project** Only	All MODBUS application data units are sent through TCP to registered port 502.
Connection to use	CONTOUSE	**Project** and **Monitoring** (for Engineer and above)	Select connection to use with redundant devices. Can be used to force a specific connection. Only available for redundant Device Types.
Connection in use	CONINUSE	No	Shows current connection is use with redundant devices. Only available for redundant Device Types.
Preferred connection to use	PREFERREDCONN	**Project** and **Monitoring** (for Operator and above)	Define the connection to use when the Connection to use value is Auto. Requests that use of specific connection be attempted. Only available for redundant Device Types.
Connection Switch Period (sec)	REDSWITCHPERIOD	**Project** and **Monitoring** (for Engineer and above)	Define redundant switch period is seconds. Only available for redundant Device Types.
Use Keep Alive	USEKEEPALIVE	**Project** Only	Enable or disable the Modbus KEEPALIVE option.

5.11.3 Slave Configuration tab parameters

The following table summarizes the parameter data you can monitor and/or configure on the **Slave Configuration** tab of the configuration form for the selected PCDI_MASTER block. Users must have an access level of *Engineer* or above to configure values in the Monitoring mode. Remember to check the **Show Parameter Names** check box, if you want to display the parameter names instead of the plain text on the form.

Plain Text	Parameter Name	User Configurable	Notes
Timeout (ms)	DEFTIMOUT	**Project and Monitoring** (for Engineer or above)	Set Modbus request timeout period in milliseconds.
Maximum Request Retries	REQRTRY	**Project and Monitoring** (for Engineer or above)	Set number of retry counts Modbus device driver attempts after a failed read or write.
Per Slave Requests	MAXUIDREQ	**Project and Monitoring** (for Engineer or above)	Set maximum number of active requests to a serial device.
Max Transactions	MAXPENDREQ	**Project and Monitoring** (for Engineer or above)	Set maximum number of transactions for PCDI_MASTER device block.
Message Delay (ms)	MSGDELAY	**Project and Monitoring** (for Engineer or above)	Set message delay time in milliseconds.
Rate (ms)	LOOPRATE	**Project and Monitoring** (for Engineer or above)	Set time between testing each end device in milliseconds.
Address	LOADADDR	**Project and Monitoring** (for Engineer or above)	Specify loop address.
Data	LOOPDATA	**Project and Monitoring** (for Engineer or above)	Represents the data value written to diagnostic request to slave devices LOOPADDR.
Type	LOOPTYPE	**Project and Monitoring** (for Engineer or above)	Set loop type operation
Test Mode	LOOPMODE	**Project and Monitoring** (for Engineer or above)	Lets you set the desired test mode to use when a diagnostic request is issued. See parameter definition for details.
Data Change	LOOPDATACHG	**Project and Monitoring** (for Engineer or above)	The data value written to a diagnostic request to slave devices.
Unit ID	UNITID[0..15]	**Project** Only	Device Index: Only available for native Modbus device.
Command	N/A	**Project** Only	List of selectable commands for native Modbus device.
Serial Device Definitions	N/A	**Project and Monitoring** (for Engineer or above)	List of Modbus RTU devices associated with Modbus Bridge: Only available with Device Type Modbus Bridge.

5.11.4 Module Statistics tab parameters

The following table summarizes the parameter data you can mostly monitor only on the **Module Statistics** tab of the configuration form for the selected PCDI_MASTER block. Remember to check the **Show Parameter Names** check box, if you want to display the parameter names instead of the plain text on the form.

Plain Text	Parameter Name	User Configurable	Notes
Alarm Enable	ALMENBSTATE	Yes	Enable or disable alarm propagation to alarm displays.
Module State	IOMSTATE	Read Only	Defines the current execution state of the associated peer device module.
In-Alarm Flag	INALM	Read Only	
Requests Sent	TOTALREQSENT	Read Only	
Responses Rcvd	REQRSPRCVD	Read Only	
Orphaned Responses	ORPHANRESPCNT	Read Only	Number of responses received for which there was no longer an outstanding request. Request may have timed-out, or has been cancelled.
Max Msgs Rcvd/sec	MAXRCVMSGPERSEC	Read Only	
Max Bytes Rcvd/sec	MAXRCVBYTESPERSEC	Read Only	
Max Msgs Sent/sec	MAXXMITMSGPERSEC	Read Only	
Max Bytes Sent/sec	MAXXMITBYTESPERSEC	Read Only	
Avg Msgs Rcvd/sec	AVGRCVMSGPERSEC	Read Only	
Avg Bytes Rcvd/sec	AVGRCVBYTESPERSEC	Read Only	
Avg Msgs Sent/sec	AVGXMITMSGPERSEC	Read Only	
Avg Bytes Sent/sec	AVGXMITBYTESPERSEC	Read Only	
Unit Index	UNITIDMON[0..15]	Read Only	Unit index (MODBUS address) for each end device.
Device Status	DEVSTS[0..15]	Read Only	Device status as determined by diagnostics.
Device Type	UIDDEVTYPEMON[0..15]	Read Only	Reads corresponding value of UIDDEVTYPE[0-15]
Honeywell Info	VENDOR[0..15]	Read Only	
Requests Sent	UIDREQSENT[0..15]	Read Only	
Responses Received	UIDREQRSPRCVD[0-15]	Read Only	
Request Retries	REQRTRYCNT[0-15]	Read Only	
Request Timeouts	TIMOUTCNT[0-15]	Read Only	
Queued Requests	QUEUEDREQCNT[0-15]	Read Only	

Plain Text	Parameter Name	User Configurable	Notes
UID Orphaned Responses	UIDORPHANRESPCNT[0-15]	Read Only	
Diag Requests	DIAGREQSENT[0-15]	Read Only	
Diag Responses	DIAGREQRSPRCVD[0-15]	Read Only	
Expected Data	UIDEXPECTDATA[0..15]	Read Only	
Received Data	UIDRCVDDATA[0..15]	Read Only	
Error Count	MBERRCNT[0..15]	Read Only	
Error Flag	MBERRFL[0..15	Read Only	
Last Error	LASTMBERR[0..15]	Read Only	
Last Error Time	LASTMBERRTIME[0-15]	Read Only	

5.11.5 Connection Statistics tab parameters

The following table summarizes the parameter data you can monitor only on the **Connection Statistics** tab of the configuration form for the selected PCDI_MASTER block. Remember to check the **Show Parameter Names** check box, if you want to display the parameter names instead of the plain text on the form.

Plain Text	Parameter Name	User Configurable	Notes
Primary			
IP Address	PRIMIPMON	Read Only	
TCP Port	PRIMTCPMON	Read Only	
Connection Status	PRIMCONNSTS	Read Only	
Successful Connects	PRIMNUMCONN	Read Only	
Conn Attempts	PRIMNUMCONNATMPT	Read Only	
Last connect time	PRIMLASTCONNTIME	Read Only	
Disconnects	PRIMNUMDISCONN[0..15]		
Last Disconnect Time	PRIMLASTDISCONNTIME[0..15]		
Error Count	PRIMERRCNT	Read Only	
Error Flag	PRIMERRFL	Read Only	When Set, Error Flag parameter indicates that Last Error is active.
Last Error	PRIMERRCODE	Read Only	
Additional Error Info	PRIMERRINFO	Read Only	
Last Error Time	PRIMLASTERRTIME	Read Only	
Secondary			
IP Address	SECIPMON	Read Only	
TCP Port	SECTCPMON	Read Only	
Connection Status	SECCONNSTS	Read Only	
Successful Connects	SECNUMCONN	Read Only	

Plain Text	Parameter Name	User Configurable	Notes
Conn Attempts	SECNUMCONNATMPT	Read Only	
Last connect time	SECLASTCONNTIME	Read Only	
Disconnects	SECNUMDISCONN[0..15]	Read Only	
Last Disconnect Time	SECLASTDISCONNTIME[0..15]	Read Only	
Error Count	SECERRCNT	Read Only	
Error Flag	SECERRFL	Read Only	When Set, Error Flag parameter indicates that Last Error is active.
Last Error	SECERRCODE	Read Only	
Additional Error Info	SECERRINFO	Read Only	
Last Error Time	SECLASTERRTIME	Read Only	

5.11.6 Channel Status tab parameters

The following table summarizes the parameter data you can monitor on the **Channel Status** tab of the configuration form for the selected PCDI_MASTER block. Remember to check the **Show Parameter Names** check box, if you want to display the parameter names instead of the plain text on the form.

Plain Text	Parameter Name	User Configurable	Notes
Assigned FB	CHANFBNAME[0..63]	Read Only	
Ch Status	CHANSTS[0..63]	Read Only	
Requests Sent	CHANREQSENT[0..63]	Read Only	
Responses	CHANREQRSPRCVD[0..63]	Read Only	
Cycle Time (ms)	CYCLETIME[0..63]	Read Only	Reports total elapsed time of the last cycle of operations in milliseconds for the request block to complete the cycle. Includes all write and read requests. **Example:** If the CM cycle time is 50ms and the function block performs single write and read, the request for write is issued at the start of the channel cycle. If the write completes before the next CM cycle starts, the write response is processed and the request for read is issued. At the next execution cycle, the channel cycle completes if the read is completed. The channel cycle time is 50ms + 50ms + read processing time (~ 1ms) = 101ms. If the request block is set to perform WriteOnChg or WriteOnDiff, the number of writes varies depending on the inputs to the request block.
Err Count	CHANMBERRCNT[0..63]	Read Only	
Err Flag	CHANMBERRFL[0..63]	Read Only	When Set, Error Flag parameter indicates that Last Error is active.

Plain Text	Parameter Name	User Configurable	Notes
Last Error	CHANLASTMBERR[0..63]	Read Only	
Err Time	CHANLASTMBERRTIME[0..63]	Read Only	

5.11.7 QVCS tab parameters

The **QVCS** tab is common to all configuration forms for tagged blocks in Control Builder. If you have a Qualification and Version Control System (QVCS) license, this tab shows current QVCS information for the selected PCDI_MASTER block. Please refer to the online help and the Qualification and Version Control System User's Guide for more information about the data on this tab.

5.11.8 Server History tab parameters

The Server **History** tab is common to all configuration forms for tagged blocks in Control Builder. The following table summarizes the parameter data you can monitor and configure on this tab of the configuration form for the selected PCDI_MASTER block.

Attention

The configuration settings you make for Server Load Options on the **System Preferences** dialog determines whether or not the data entered on the **Server History** tab is loaded to the Experion Server. See the *Control Building Guide* for information about setting system preferences.

Plain Text	Parameter Name	User Configurable	Notes
Access Levels			
Control Level	SCANCTRLLVL	Yes	Indicates Server control level to be associated with this function.
History Configuration			
Number of History Parameters	HIST.NUMPARAMS	Yes	Defines number of history parameters to be included in History Configuration table.
Parameter	HIST.PARAM	Yes	Valid parameter name for a parameter associated with the given point that is to be collected and stored as historical data at predetermined intervals.
Description		No	Provides a brief description of the entered parameter.
FAST	HIST.FAST	Yes	Select the Fast type of history collection.
STD	HIST.STD	Yes	Select the Standard type of history collection.
EXTD	HIST.EXTD	Yes	Select the Extended type of history collection.
EXC	HIST.EXC	Yes(**Station** only)	Select the Exception type of history collection.
Gating Parameter	HIST.GATEPARAM	Yes	Optional gating parameter to define conditions under which data for this parameter should be collected.

Plain Text	Parameter Name	User Configurable	Notes
Gate State	HIST.GATEVALUE	Yes	Defines gate state for configured gating parameter.
Create New or Edit Existing Server Scripts (Button)		N/A	Launch the Server scripting configuration utility.

5.11.9 Server Displays tab parameters

The **Server Displays** tab is common to all configuration forms for tagged blocks in Control Builder. The following table summarizes the parameter data you can monitor and configure on this tab of the configuration form for the selected PCDI_MASTER block.

Attention

The configuration settings you make for Server Load Options on the **System Preferences** dialog determines whether or not the data entered on the **Server Displays** tab is loaded to the Experion Server. See the *Control Building Guide* for information about setting system preferences.

Plain Text	Parameter Name	User Configurable	Notes
Point Detail Display	SCANPNTDTL	Yes	By default, a Display template is already entered into Point Detail Display box (for example, Sysdtlmbtcpdevicea). This template can be used for creating your own display or it can be used as is, provided that your function block name matches name built into detail display that is supplied as a template.
Group Detail Display	SCANGRPDTL	Yes	By default, a Display template is already entered into the Group Detail Display box (for example, Sysdtlmbtcpdevicea_fp). This template can be used for creating your own display or it can be used as is, provided that your function block name matches name built into detail display that is supplied as a template
Associated Display	SCANASSOCDSP	Yes	Name of the Server display to be associated with this function block.
Trends			
Number of Trends	TREND.NUMPARAMS	Yes	Defines the number of trend parameters to be included in the Trends Configuration table.
Trend #		Yes	Defines Trend number to be associated with this trend parameter
Pen		Yes	Defines color of pen that will be used to trace assigned parameter on Station Trend display.
Trend Parameter		Yes	Valid parameter name for a parameter associated with given point that is configured for history collection.
Description		No	Provides a brief description of the entered parameter.
Groups			

Plain Text	Parameter Name	User Configurable	Notes
Number of Groups	GROUP.NUMPARAMS	Yes	Defines the number of group parameters to be included in Groups Configuration table.
Group #		Yes	Defines Group number to be associated with this group parameter.
Pos #		Yes	Defines number of position configured parameter will occupy in the Station Group display.
Group Parameter		Yes	Valid parameter name for a parameter associated with the given point that is configured in the system.
Description		No	Provides a brief description of the entered parameter.

5.11.10 Control Confirmation tab parameters

The **Control Confirmation** tab is common to all configuration forms for tagged blocks in Control Builder. If you have an optional Electronic Signature license, you can configure electronic signature information for the tagged block through this tab on the block's configuration form in Control Builder. Please refer to the online help and the *Server and Client Configuration Guide* for information about the data on this tab.

The Electronic Signature function aligns with the identical Electronic Signatures function that is initiated through Quick Builder and Station for Server points. When this block is loaded to a controller, its control confirmation configuration (electronic signatures) is also loaded to the Server. This means you can view the control confirmation configuration for this tagged object in Station and also make changes to it. If you make changes through Station, you must initiate an **Upload** or **Upload with Contents** function through the **Controller** menu in Control Builder for the object in the **Monitoring** tab to synchronize changes in the Engineering Repository Database (ERDB).

5.11.11 Identification tab parameters

The Identification tab is common to all configuration forms for blocks in Control Builder. The following table summarizes the parameter data you can monitor and configure on this tab of the configuration form for the selected PCDI_MASTER block.

Plain Text	Parameter Name	User Configurable	Notes
Name	NAME	Yes	Unique block name consisting of up to 16 characters to identify the block. At least one character in the name must be a letter (A-Z).
Description	DESC	Yes	Descriptive text appears on detail and group displays to uniquely describe this particular function block
Block Comment 1	BLCKCOMMENT1	Yes	Comment to be associated with this block consisting of up to 40 characters.
Block Comment 2	BLCKCOMMENT2	Yes	Comment to be associated with this block consisting of up to 40 characters.
Block Comment 3	BLCKCOMMENT3	Yes	Comment to be associated with this block consisting of up to 40 characters.

Plain Text	Parameter Name	User Configurable	Notes
Block Comment 4	BLCKCOMMENT4	Yes	Comment to be associated with this block consisting of up to 40 characters.
Library		No	Identifies Control Builder Library that is source of template.
System Template		No	Identifies System Template that is source for this block.
Base Template		No	Identifies Base Template that is used for this block.
Created By	CREATEDBY	No	Identifies user who created block, if operator security is implemented. Otherwise, may just show Default login.
Date Created	DATECREATED	No	Shows date and time template was created. If this block is in Version Control System, shows date and time initial version of template was created.
Last Modified By	MODIFIEDBY	No	Identifies user who made last modifications to block, if operator security is implemented. Otherwise, may just show default login. If this block is in Version Control System, modifications apply to last version of block.
Date Last Modified	VERSIONDATE	No	Shows date and time last modification was made to block's configuration. If this block is in Version Control System, modification date and time applies to last version of block.

5.12 Array Request Channel Block properties form reference

Related topics

“Main tab parameters” on page 75
“Configuration tab parameters” on page 75
“Scaling/Substitution tab parameters” on page 77
“Status/Data tab parameters” on page 78
“Identification tab parameters” on page 78
“Dependencies tab parameters” on page 79
“Block pins tab parameters” on page 79
“Configuration parameters tab parameters” on page 79
“Monitoring parameters tab parameters” on page 79
“Block preferences tab parameters” on page 79
“Template defining tab parameters” on page 79

5.12.1 Main tab parameters

The following table summarizes the parameter data you can monitor and/or configure on the **Main** tab of the configuration form for the selected PCDIFLAGARRCH, PCDINUMARRCH, or PCDITEXTARRCH block. Remember to check the **Show Parameter Names** check box, if you want to display the parameter names instead of the plain text on the form.

Plain Text	Parameter Name	User Configurable	Notes
Request Block Name	NAME	**Project** Only	System assigned or user configured unique name. Consisting of up to 15 characters and at least one character must be a letter (A-Z).
Execution Order in CM	ORDERINCM	**Project** Only	Determines the execution order of function blocks within a Control Module (CM)
Master Block Name	N/A	**Project** Only	Select associated PCDI_MASTER block.

5.12.2 Configuration tab parameters

The following table summarizes the parameter data you can monitor and/or configure on the **Configuration** tab of the configuration form for the selected PCDIFLAGARRCH, PCDINUMARRCH, or PCDITEXTARRCH block. Remember to check the **Show Parameter Names** check box, if you want to display the parameter names instead of the plain text on the form.

Plain Text	Parameter Name	User Configurable	Notes
Access Lock	ACCLOCK	**Project** Only	Select access level for block.

Plain Text	Parameter Name	User Configurable	Notes
Auto Trigger Mode	AUTOTRIGGER	**Project** Only	Enable or disable the auto trigger mode of request block execution. In the *triggered* mode, the block submits a request on the positive transition of the input trigger flag. In the *auto triggered* mode, the DONEFL is internally looped back to the trigger to provide for the fastest update rate possible. In this mode, the block will be capable of refreshing data at the same rate as the execution cycle of the containing CM, but limited by the response time of the addressed device.
Auto Trigger Period (ms)	AUTOTRIGGERPERI OD	**Project** Only	Used to delay the beginning of the following auto triggered block execution. Auto triggered execution will resume in the CM cycle following expiration of the auto trigger period. If the auto trigger period has expired before the DONEFL parameter transitions to true, the block will be auto triggered immediately. The default AUTOTRIGGERPERIOD value is 0 milliseconds, which results in the block being triggered as quickly as possible.
Device Address	DEVADDR	**Project** Only	Specify applicable MODBUS device address.
Starting Element Index	STARTINDEX	**Project** Only	Set applicable start index value for given Array Request Channel block
Long	CONVTOLONGI	**Project** Only	Applicable to PCDINUMARRCH block only. Enable or disable convert to long function.
ASCII Conversion	CONVTOASCII	**Project** Only	Applicable to PCDITEXTARRCH block only. Enable or disable convert text to ASCII function.
Number of Flag Values	NFLAG	**Project** Only	Applicable to PCDIFLAGARRCH block only. Define number of flags to read. Range is 1 to 1968.
Number of Numeric Values	NNUMERIC	**Project** Only	Applicable to PCDINUMARRCH block only. Define number of numeric values to read. Valid values are actually dependent upon the STARTINDEX.

Plain Text	Parameter Name	User Configurable	Notes
Number of String Values	NSTRING	**Project** Only	Applicable to PCDITEXTARRCH block only. Define number of string values to read. The number of strings times the number of characters per string may not exceed 120 Unicode characters or 240 ASCII characters.
Char Length of String Values	STRLEN	**Project** Only	Applicable to PCDITEXTARRCH block only. Define number of characters per string The number of strings times the number of characters per string may not exceed 120 Unicode characters or 240 ASCII characters.
Read/Write Option	WRITEOPT	**Project** Only	Select read/write setting for block.

5.12.3 Scaling/Substitution tab parameters

The following table summarizes the parameter data you can monitor and/or configure on the **Scaling/ Substitution** or **Substitution** tab of the configuration form for the selected PCDINUMARRCH, or PCDIFLAGARRCH or PCDITEXTARRCH block. Remember to check the **Show Parameter Names** check box, if you want to display the parameter names instead of the plain text on the form.

Plain Text	Parameter Name	User Configurable	Notes
Default Substitution Type	SUBVALTYPE	**Project** Only	Select what value to substitute when error occurs.
Default Substitution Value	SUBVAL	**Project** Only	Applicable to PCDINUMARRCH and PCDITEXTARRCH blocks only. Set value to use when error occurs.
Set to ON on error	SUBVAL	**Project** Only	Applicable to PCDIFLAGARRCH block only. Enable or disable set to ON on error function.
PV High Range (in Engineering Units)	PVEUHI	**Project** Only	Applicable to PCDINUMARRCH block only. Specify high range value in Engineering Units.
PV Low Range (in Engineering Units)	PVEULO	**Project** Only	Applicable to PCDINUMARRCH block only. Specify low range value in Engineering Units.
PV Raw High Range	PVRAWHI	**Project** Only	Applicable to PCDINUMARRCH block only. PV raw high range value.
PV Raw Low Range	PVRAWLO	**Project** Only	Applicable to PCDINUMARRCH block only. PV raw low range value.

Plain Text	Parameter Name	User Configurable	Notes
Engineering Unit Description	EUDESC	**Project** Only	Applicable to PCDINUMARRCH block only. Provide description of applicable Engineering Unit.
Substitution Type	ELEMSUBVALTYPE	**Project** Only	Substitution on error is supported with common SUBVALTYPE and SUBVAL parameters that may be overridden with parameters ELEMSUBVALTYPE and ELEMSUBVAL on per element basis.
Substitution Value	ELEMSUBVAL	**Project** Only	See row above.

5.12.4 Status/Data tab parameters

The following table summarizes the parameter data you can monitor on the **Status/Data** tab of the configuration form for the selected PCDIFLAGARRCH, PCDINUMARRCH, or PCDITEXTARRCH block. Remember to check the **Show Parameter Names** check box, if you want to display the parameter names instead of the plain text on the form.

Plain Text	Parameter Name	User Configurable	Notes
Ch Status	CHANSTS	**Monitoring** Only	See parameter definition for details.
Error Flag	ERRFL	**Monitoring** Only	Enable or disable error flag function. When set, indicates that ERRCODE is currently active.
Last Error	ERRCODE	**Monitoring** Only	See parameter definition for details.
Bad PV Flag	BADPVFL	**Monitoring** Only	
Time of Last Error	LASTERRTIME	No	
Process Value	PVFL	**Monitoring** Only	Applicable to PCDIFLAGARRCH block only.
Process Value	PV	**Monitoring** Only	Applicable to PCDINUMARRCH block only.
Process Value Raw	PVRAW	**Monitoring** Only	Applicable to PCDINUMARRCH block only.
Process Value	STR[1..16	**Monitoring** Only	Applicable to PCDITEXTARRCH block only.

5.12.5 Identification tab parameters

The **Identification** tab is common to all configuration forms for basic blocks in Control Builder. The following table summarizes the parameter data you can monitor and configure on this tab of the configuration form for the selected Array Request Channel block.

Plain Text	Parameter Name	User Configurable	Notes
Name	NAME	Yes	Unique block name consisting of up to 15 characters to identify the block. At least one character in the name must be a letter (A-Z).

Plain Text	Parameter Name	User Configurable	Notes
Block Comment 1	BLCKCOMMENT1	Yes	Comment to be associated with this block consisting of up to 40 characters.
Block Comment 2	BLCKCOMMENT2	Yes	Comment to be associated with this block consisting of up to 40 characters.
Block Comment 3	BLCKCOMMENT3	Yes	Comment to be associated with this block consisting of up to 40 characters.
Block Comment 4	BLCKCOMMENT4	Yes	Comment to be associated with this block consisting of up to 40 characters.
Library		No	Identifies Control Builder Library that is source of template.
System Template		No	Identifies System Template that is source for this block.
Base Template		No	Identifies Base Template that is used for this block.

5.12.6 Dependencies tab parameters

The **Dependencies** tab is common to all configuration forms for basic blocks in Control Builder. Please refer to the online help and the Control Builder User's Guide for more information about the data on this tab.

5.12.7 Block pins tab parameters

The **Block Pins** tab is common to all configuration forms for basic blocks in Control Builder. Please refer to the online help and the Control Builder User's Guide for more information about the data on this tab.

5.12.8 Configuration parameters tab parameters

The **Configuration Parameters** tab is common to all configuration forms for basic blocks in Control Builder. Please refer to the online help and the Control Builder User's Guide for more information about the data on this tab.

5.12.9 Monitoring parameters tab parameters

The Monitoring Parameters tab is common to all configuration forms for basic blocks in Control Builder. Please refer to the online help and the Control Builder User's Guide for more information about the data on this tab.

5.12.10 Block preferences tab parameters

The **Block Preferences** tab is common to all configuration forms for basic blocks in Control Builder. Please refer to the online help and the Control Builder User's Guide for more information about the data on this tab.

5.12.11 Template defining tab parameters

The **Template Defining** tab is common to all configuration forms for basic blocks in Control Builder. Please refer to the online help and the Control Builder User's Guide for more information about the data on this tab.

6 Peer Control Data Interface operation

Related topics

“Monitoring Peer Control Data Interface functions through Station displays” on page 82
“Monitoring PCDI Functions through the Monitoring tab in Control Builder” on page 90
“Initiating Switchover of Redundant Devices” on page 96
“Checking license details” on page 97
“Response to C300 RAM retention restart” on page 98
“PCDI support for checkpoint save/restore functions” on page 99

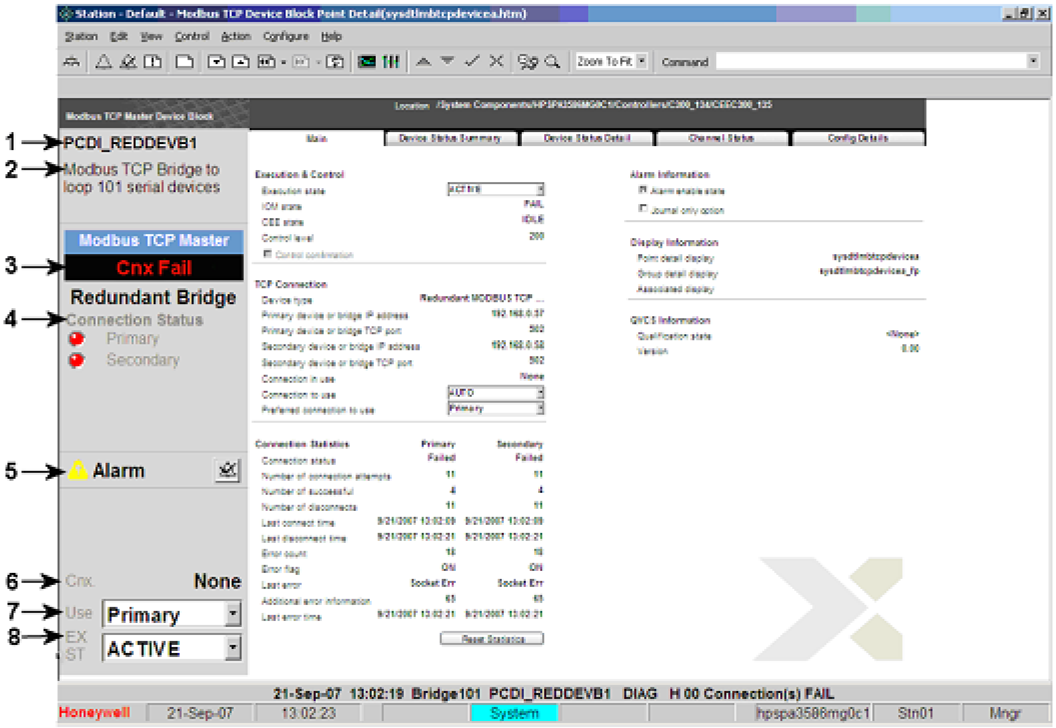
6.1 Monitoring Peer Control Data Interface functions through Station displays

The Experion Server Station application includes pre-configured Detail displays for the PCDI_MASTER function blocks. These displays are the default entries for the Point Detail Page parameter on the Server Displays tab of the configuration form. Once you establish communications with a peer device, you can begin monitoring the status of any component that has been loaded as part of a Control Strategy to an Experion Controller with points registered in the Experion Server. The Detail displays let you quickly view the component's current state, fault status, and pertinent configuration data.

Please refer to the *Operator's Guide* for detailed information about calling up, navigating, and viewing displays in Station.

6.1.1 Graphical example of PCDI_MASTER block detail display

The following illustration shows the faceplate for a PCDI master device displaying its component state. See the following tables for description of illustration callouts and peer device text messages.



Station - Default - Modbus TCP Device Block Point Detail(sysdtlmbtcpdevicea.htm)

Location /System Components/HPSPA3586MG0C1/Controllers/C300_134/CEEC300_135

Modbus TCP Master Device Block

PCDI_REDDEVB1

Modbus TCP Bridge to loop 101 serial devices

Modbus TCP Master

Cnx Fail

Redundant Bridge

Connection Status

● Primary

● Secondary

Alarm

Cnx None

Use Primary

EX ST ACTIVE

Execution & Control
Execution state ACTIVE
IOM state FAIL
CEE state IDLE
Control level 200
□ Control confirmation

TCP Connection
Device type Redundant MODBUS TCP ...
Primary device or bridge IP address 192.168.0.37
Primary device or bridge TCP port 502
Secondary device or bridge IP address 192.168.0.38
Secondary device or bridge TCP port 502
Connection in use None
Connection to use AUTO
Preferred connection to use Primary

Connection Statistics	Primary	Secondary
Connection status	Failed	Failed
Number of connection attempts	11	11
Number of successful	4	4
Number of disconnects	11	11
Last connect time	9/21/2007 13:02:09	9/21/2007 13:02:09
Last disconnect time	9/21/2007 13:02:21	9/21/2007 13:02:21
Error count	18	18
Error flag	ON	ON
Last error	Socket Err	Socket Err
Additional error information	65	65
Last error time	9/21/2007 13:02:21	9/21/2007 13:02:21

Alarm Information
☒ Alarm enable state
□ Journal only option

Display Information
Point detail display sysdtlmbtcpdevicea
Group detail display sysdtlmbtcpdevicea_fp
Associated display

QVCS Information
Qualification state <None>
Version 0.00

21-Sep-07 13:02:19 Bridge101 PCDI_REDDEVB1 DIAG H 00 Connection(s) FAIL

Honeywell 21-Sep-07 13:02:23 System hpspa3586mg0c1 Stn01 Mngr

Figure 5: Sample PCDI Master Device Detail Display

Table 3: Description of Figure 5 Callouts.

Callout	Description
1	Module name configured through Control Builder.
2	Module description configured through Control Builder.
3	Peer device text indicating current general status of functions.
4	LEDs show current connection status.
5	Shows any alarm condition. Click button to acknowledge alarm.

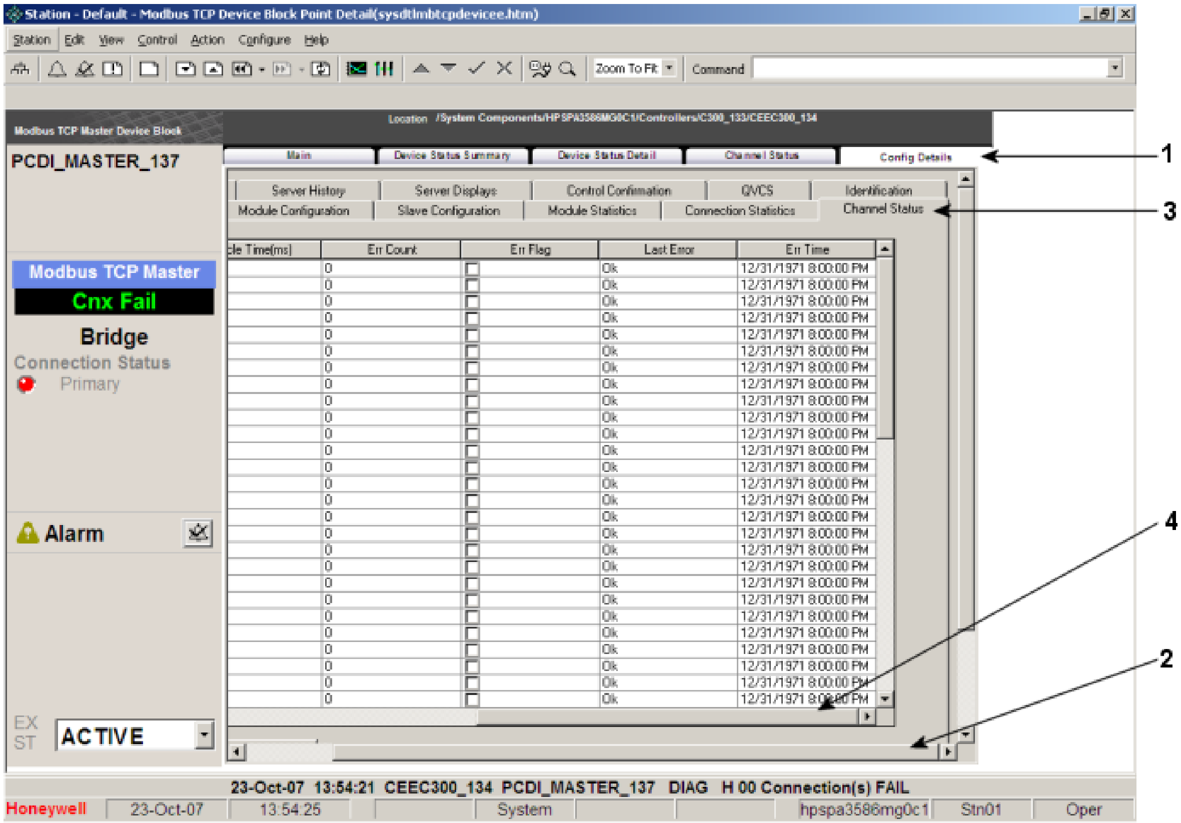
Callout	Description
6	Identifies the connection in use.
7	Identifies the preferred connection to use.
8	Shows the current execution status of the device.

Table 4: Table Peer Device Text Description

If text is . . .	**Then, for a single or pair of redundant Safety Managers, Modbus TCP Bridges, or Modbus TCP devices it means . . .**
Inactive	PCDI_MASTER Block is Inactive
Initializing	Connection(s) in transition
Cnx Fail	Connection(s) failed.
Cnx Offline	No Connections available. (That is: CONTOUSE = NONE.)
If text is . . .	**Then, for a single or pair of redundant Modbus TCP Bridges, or Modbus TCP devices it means . . .**
Cnx OK DV Init	Connection(s) OK, all peer devices/Modbus RTU devices in transition
Cnx OK DV OK	Connection(s) OK and all peer devices/Modbus RTU devices OK.
Cnx OK DV PF	Connection(s) OK and at least one peer device/Modbus RTU device failed (That is: peer devices/Modbus RTU devices partially failed).
Cnx OK DV POK	Connection(s) OK, at least one peer device/Modbus RTU device OK and no peer devices/Modbus RTU devices failed (That is: peer devices/Modbus RTU devices partially OK).
Cnx OK DV Fail	Connection(s) OK and all peer devices/Modbus RTU devices failed.
If text is . . .	**Then, for a pair of redundant Modbus TCP Bridges or Modbus TCP devices, it means . . .**
Cnx Marg DV Init	One connection failed and the alternate connection is in use, all peer devices/ Modbus RTU devices in transition
Cnx Marg DV OK	One connection failed and the alternate connection is in use, all peer devices/ Modbus RTU devices OK.
Cnx Marg DV PF	One connection failed and the alternate connection is in use and at least one peer device/Modbus RTU device failed (That is: peer devices/Modbus RTU devices partially failed).
Cnx Marg DV POK	One connection failed and the alternate connection is in use and at least one peer device/Modbus RTU device OK and no peer devices/Modbus RTU devices failed (That is: peer devices/Modbus RTU devices partially OK).
Cnx Marg DV Fail	One connection failed and the alternate connection is in use and all peer devices/ Modbus RTU devices failed.
If text is . . .	**Then, for a single or pair of redundant Safety Managers, it means . . .**
Cnx OK SM Init	Connection(s) OK, all Safety Manager devices in transition
Cnx OK SM OK	Connection(s) OK and all Safety Manager devices OK.
Cnx OK SM PF	Connection(s) OK and at least one Safety Manager device failed (That is: Safety Manager devices partially failed).
Cnx OK SM POK	Connection(s) OK, at least one Safety Manager device OK and no Safety Manager devices idle or failed (That is: Safety Manager devices partially OK).
Cnx OK SM Fail	Connection(s) OK and all Safety Manager devices failed.
Cnx OK SM Idle	Connection(s) OK, at least one Safety Manager device idle and no Safety Manager device failed.
If text is . . .	**Then, for a pair of redundant Safety Managers, it means . . .**

Cnx Marg SM Init	One connection failed and the alternate connection is in use, all Safety Manager devices in transition
Cnx Marg SM OK	One connection failed and the alternate connection is in use, all Safety Manager devices OK.
Cnx Marg SM PF	One connection failed and the alternate connection is in use and at least one Safety Manager device failed (That is: Safety Manager devices partially failed).
Cnx Marg SM POK	One connection failed and the alternate connection is in use and at least one Safety Manager device OK and no Safety Manager devices idle or failed (That is: Safety Manager devices partially OK).
Cnx Marg SM Fail	One connection failed and the alternate connection is in use and all Safety Manager devices failed.
Cnx Marg SM Idle	One connection failed and the alternate connection is in use, at least one Safety Manager device idle and no Safety Manager device failed.

6.1.2 Graphical example of viewing Last Error on Channel Status tab of PCDI_MASTER block in Station



le Time(ms)	Err Count	Err Flag	Last Error	Err Time
	0	□	Ok	12/31/1971 8:00:00 PM
	0	□	Ok	12/31/1971 8:00:00 PM
	0	□	Ok	12/31/1971 8:00:00 PM
	0	□	Ok	12/31/1971 8:00:00 PM
	0	□	Ok	12/31/1971 8:00:00 PM
	0	□	Ok	12/31/1971 8:00:00 PM
	0	□	Ok	12/31/1971 8:00:00 PM
	0	□	Ok	12/31/1971 8:00:00 PM
	0	□	Ok	12/31/1971 8:00:00 PM
	0	□	Ok	12/31/1971 8:00:00 PM
	0	□	Ok	12/31/1971 8:00:00 PM
	0	□	Ok	12/31/1971 8:00:00 PM
	0	□	Ok	12/31/1971 8:00:00 PM
	0	□	Ok	12/31/1971 8:00:00 PM
	0	□	Ok	12/31/1971 8:00:00 PM
	0	□	Ok	12/31/1971 8:00:00 PM
	0	□	Ok	12/31/1971 8:00:00 PM
	0	□	Ok	12/31/1971 8:00:00 PM
	0	□	Ok	12/31/1971 8:00:00 PM
	0	□	Ok	12/31/1971 8:00:00 PM
	0	□	Ok	12/31/1971 8:00:00 PM
	0	□	Ok	12/31/1971 8:00:00 PM
	0	□	Ok	12/31/1971 8:00:00 PM
	0	□	Ok	12/31/1971 8:00:00 PM
	0	□	Ok	12/31/1971 8:00:00 PM
	0	□	Ok	12/31/1971 8:00:00 PM
	0	□	Ok	12/31/1971 8:00:00 PM

23-Oct-07 13:54:21 CEEC300_134 PCDI_MASTER_137 DIAG H 00 Connection(s) FAIL

Honeywell | 23-Oct-07 | 13:54:25 | System | hpspa3586mg0c1 | Stn01 | Oper

Figure 6: Example View of Last Error on Channel Status Tab in Station

1 Click the **Config Details** tab on Detail Display.

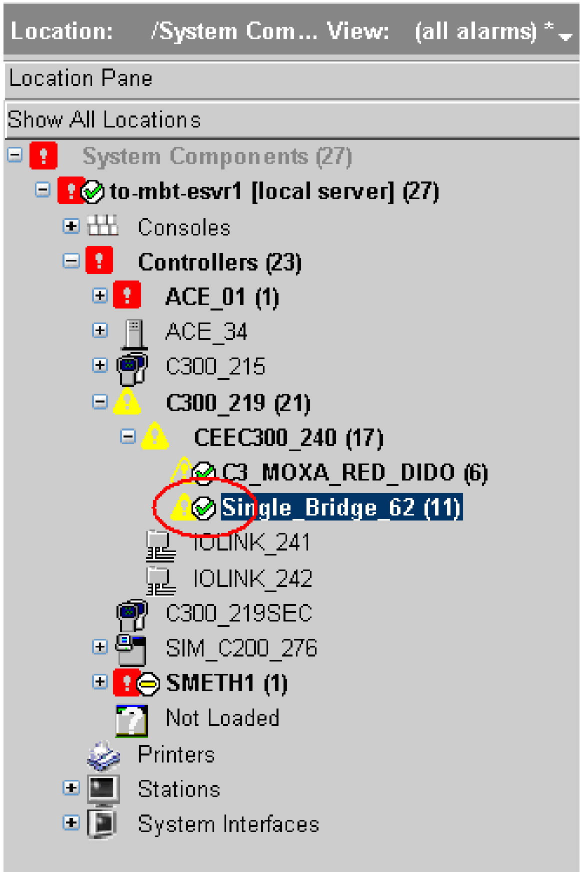
2 Click scroll bar for tab and move to right side.

3 Click the **Channel Status** tab.

4 Click the scroll bar on the tab and move to the right side to expose the **Last Error** column.

6.1.3 Graphical example of PCDI_MASTER block in system component tree

The following illustration shows a PCDI master device being displayed on the system component tree in Station. See the following table for list and description of possible status icon conditions.



Location: /System Com... View: (all alarms) *

Location Pane

Show All Locations

```
System Components (27)
  to-mbt-esvr1 [local server] (27)
    Consoles
    Controllers (23)
      ACE_01 (1)
      ACE_34
      C300_215
      C300_219 (21)
        CEEC300_240 (17)
          C3_MOXA_RED_DIDO (6)
          Single_Bridge_62 (11)
        IOLINK_241
        IOLINK_242
      C300_219SEC
      SIM_C200_276
      SMETH1 (1)
      Not Loaded
    Printers
    Stations
    System Interfaces
```

Figure 7: Typical System Component Tree View

Table 5: Table Status Icon Description

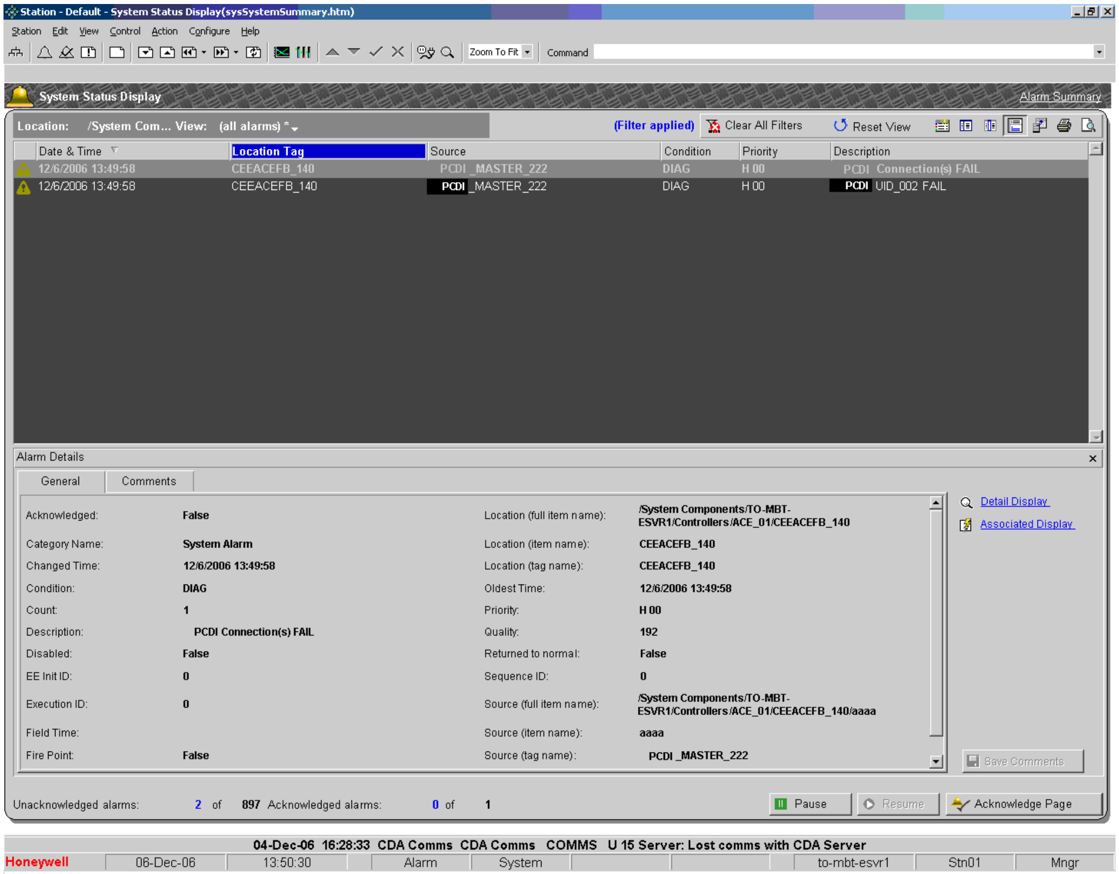
If Status Icon is . . .	Then, It Means
✓	Connection(s) OK, all peer devices/Modbus RTU devices OK
⊖	Connection(s) marginal or peer devices/Modbus RTU devices marginal
⊗	Connection(s) failed or any peer devices/Modbus RTU devices failed
◉	Master Block is Inactive

6.1.4 Using Station System Status display

Like the Detail displays, the System Status Display supports the integration of PCDI Master Device block generated notifications and events. It is integrated with Experion component data and is for the most part self-explanatory. PCDI channel blocks will report channel alarms through the Device Block they are assigned to. Channel alarms will be returned to normal if alarm condition no longer exists, the channel block state is changed to inactive, or the channel block is deleted. Use this display to get a quick review of recent actions that have been initiated within the system.

6.1.5 Graphical example of PCDI_MASTER block in Alarm pane of System Status Display

The following illustration shows alarm events for PCDI master device. See the following table for list and description of possible Peer Control Data Interface event subtypes in the Experion Notification System based on a per PCDI Master Device block basis.



Location: /System Com... View: (all alarms) * (Filter applied)

Date & Time	Location Tag	Source	Condition	Priority	Description
12/6/2006 13:49:58	CEEACEFB_140	PCDI _MASTER_222	DIAG	H 00	PCDI Connection(s) FAIL
12/6/2006 13:49:58	CEEACEFB_140	PCDI _MASTER_222	DIAG	H 00	PCDI UID_002 FAIL

Alarm Details

Acknowledged:	False	Location (full item name):	/System Components/TO-MBT-ESVR1/Controllers/ACE_01/CEEACEFB_140
Category Name:	System Alarm	Location (item name):	CEEACEFB_140
Changed Time:	12/6/2006 13:49:58	Location (tag name):	CEEACEFB_140
Condition:	DIAG	Oldest Time:	12/6/2006 13:49:58
Count:	1	Priority:	H 00
Description:	PCDI Connection(s) FAIL	Quality:	192
Disabled:	False	Returned to normal:	False
EE Init ID:	0	Sequence ID:	0
Execution ID:	0	Source (full item name):	/System Components/TO-MBT-ESVR1/Controllers/ACE_01/CEEACEFB_140/aaaa
Field Time:		Source (item name):	aaaa
Fire Point:	False	Source (tag name):	PCDI _MASTER_222

Unacknowledged alarms: 2 of 897 Acknowledged alarms: 0 of 1

04-Dec-06 16:28:33 CDA Comms CDA Comms COMMS U 15 Server: Lost comms with CDA Server

Honeywell	06-Dec-06	13:50:30	Alarm	System			to-mbt-esvr1	Stn01	Mngr

Figure 8: System Status Display Alarm Pane

Table 6: Peer Control Data Interface Event Subtypes

Peer Control Data Interface Event Subtype	Cond	Prty	Auxiliary Descriptor	Event Description
Connection Fail	DIAG	H	Connection(s) FAIL	For a single Modbus device, an alarm is generated when a single Modbus device connection has failed. For a redundant pair of Modbus devices, an alarm is generated when both connections of a redundant pair of Modbus devices have failed.
Connection Warning	DIAG	H	Connection Warning	For a redundant pair of Modbus devices, an alarm is generated when a single connection has failed and the alternate connection is in use.
Single Device FAIL	DIAG	H	Single Device FAIL	For a single MODBUS TCP Device, an alarm is generated when the single connection is OK, but the single MODBUS TCP Device has failed.
Primary Device FAIL	DIAG	H	Primary Device FAIL	For a redundant MODBUS TCP Device, an alarm is generated when the primary connection is OK, but the primary MODBUS TCP Device has failed.
Secondary Device FAIL	DIAG	H	Secondary Device FAIL	For a redundant MODBUS TCP Device, an alarm is generated when the secondary connection is OK, but the secondary MODBUS TCP Device has failed.
Primary Connection is forced	DIAG	H	Secondary Offline	For a redundant pair of Modbus devices, an alarm is generated when CONTOUSE is set to Primary.
Secondary Connection is forced in use	DIAG	H	Primary Offline	For a redundant pair of Modbus devices, an alarm is generated when CONTOUSE is set to Secondary.
Primary and secondary connections are forced offline	DIAG	H	No Connections Available	For a single Modbus device or a redundant pair of Modbus devices, an alarm is generated when CONTOUSE is set to NONE.
Primary All UID FAIL	DIAG	H	Primary All UID FAIL	For a single MODBUS TCP Bridge, an alarm is generated when all Modbus RTU devices on the primary connection have failed. For a redundant MODBUS TCP Bridge, an alarm is generated when all Modbus RTU devices on primary connection have failed, CONINUSE is switched to Secondary, the alarm will not return to normal until CONINUSE is switched back to Primary again and alarm condition has been cleared.

Peer Control Data Interface Event Subtype	Cond	Prty	Auxiliary Descriptor	Event Description
Secondary All UID FAIL	DIAG	H	Secondary All UID FAIL	For a redundant MODBUS TCP Bridge, an alarm is generated when all Modbus RTU devices on secondary connection have failed, CONINUSE is switched to Primary, the alarm will not return to normal until CONINUSE is switched back to Secondary again and alarm condition has been cleared.
Single Safety Manager FAIL	DIAG	H	Single SM FAIL	For a single Safety Manager, an alarm is generated when the single connection is OK, but the single Safety Manager has failed.
Primary Safety Manager FAIL	DIAG	H	Primary SM FAIL	For a redundant Safety Manager, an alarm is generated when the primary connection is OK, but the primary Safety Manager has failed.
Secondary Safety Manager FAIL	DIAG	H	Secondary SM FAIL	For a redundant Safety Manager, an alarm is generated when the secondary connection is OK, but the secondary Safety Manager has failed.
Single Safety Manager IDLE	DIAG	H	Single SM IDLE	For a single Safety Manager, an alarm is generated when the single Safety Manager is IDLE.
Primary Safety Manager IDLE	DIAG	H	Primary SM IDLE	For a redundant Safety Manager, an alarm is generated when the primary Safety Manager is IDLE.
Secondary Safety Manager IDLE	DIAG	H	Secondary SM IDLE	For a redundant Safety Manager, an alarm is generated when the secondary Safety Manager is IDLE.
Serial Device [0..15] Fail	DIAG	H	UID xxx FAIL	For a single MODBUS TCP Bridge or a redundant MODBUS TCP Bridge with up to 16 Modbus RTU devices, an alarm is generated when a Modbus RTU device has failed. (xxx is Modbus TCP Bridge Modbus RTU device Unit ID, range 1..247 and 255)
Serial Device [0..15] Disabled	DIAG	H	UID xxx Disabled	For a single MODBUS TCP Bridge or a redundant MODBUS TCP Bridge with up to 16 Modbus RTU devices, an alarm is generated when a Modbus RTU device has been disabled. (xxx is Modbus TCP Bridge Modbus RTU device Unit ID, range 1..247 and 255)
Channel Fail [0..63]	DIAG	H	Channel xxFAIL	For a single Modbus device or a redundant pair of Modbus devices, an alarm is generated when an assigned channel has failed. (xx is assigned channel number, range 0..63).

Legend:

- single Modbus device = single MODBUS TCP Bridge, single MODBUS TCP Device or Safety Manager
- redundant pair of Modbus devices = redundant MODBUS TCP Bridge, redundant MODBUS TCP Device, or redundant Safety Manager

Refer to the *Operator's Guide* for more information about using the Station interface.

6.2 Monitoring PCDI Functions through the Monitoring tab in Control Builder

Related topics

“Activating/inactivating PCDI_MASTER device” on page 90

“PCDI_MASTER device block icon appearances” on page 90

“Control Module block icon appearances” on page 93

“Monitoring/Interacting with given component/block” on page 93

“Monitoring PCDI related statistics through C300 block in Monitoring mode” on page 94

“Monitoring PCDI related statistics through CEEC300 block in Monitoring mode” on page 94

6.2.1 Activating/inactivating PCDI_MASTER device

Use the following procedure to change the state of a PCDI_MASTER device through the **Monitoring** tab in Control Builder.

Activating a PCDI_MASTER device generates a state event change and resumes the module's normal periodic processing.

Inactivating a PCDI_MASTER device generates a state event change; blocks channel access to flag, numeric, and string data from the device; and sets channel depending on SUBVALTYPE, as well as setting associated BADPVFL.

Prerequisites

- You have logged on with sufficient privileges to make changes in a control strategy.
- You have loaded a control strategy to the Controller and can view the PCDI_MASTER device through the **Monitoring** tab of Control Builder.

To activate or inactivate a PCDI_MASTER device

1. Right-click the PCDI_MASTER device icon (green) and click **Inactivate>Selected Item(s)** on the menu.
   Or,

   Right click the PCDI_MASTER device icon (blue) and click the **Activate>Selected Item(s)** on the menu.
2. Click the **Yes** button to confirm the action in the **Change State** dialog.
3. Wait for PCDI_MASTER device icon to turn blue/inactive.
   Or,

   Wait for PCDI_MASTER device icon to turn green/active.
4. This completes the procedure. Go to the next section.

6.2.2 PCDI_MASTER device block icon appearances

The following table summarizes the various appearances that a PCDI_MASTER device block icon in Control Builder can assume based on configuration, view, and current PCDI_MASTER device operating state and status.

If Icon is . . .	Then, it represents . . .	And, Module State (IOMSTATE)* Is. . .
Project Tab		
(gray)	PCDI_MASTER non-redundant bridge configured for operation in Project.	N/A
(gray)	PCDI_MASTER non-redundant MODBUS TCP device configured for operation in Project.	N/A
(gray)	PCDI_MASTER non-redundant Safety Manager configured for operation in Project.	N/A
(gray)	PCDI_MASTER redundant bridges configured for operation in Project.	N/A
(gray)	PCDI_MASTER redundant MODBUS TCP devices configured for operation in Project.	N/A
(gray)	PCDI_MASTER redundant Safety Managers configured for operation in Project.	N/A
Monitoring Tab		
↓ (gray)	PCDI_MASTER non-redundant bridge is loaded, but not monitoring	N/A
↓ (blue)	PCDI_MASTER non-redundant MODBUS TCP device is loaded, but not monitoring	N/A
↓ (green)	PCDI_MASTER non-redundant Safety Manager is loaded, but not monitoring	N/A
↓ (gray)	PCDI_MASTER redundant bridges are loaded, but not monitoring	N/A
↓ (gray)	PCDI_MASTER redundant MODBUS TCP devices are loaded, but not monitoring	N/A
↓ (gray)	PCDI_MASTER redundant Safety Managers are loaded, but not monitoring	N/A
(blue)	PCDI_MASTER non-redundant bridge is inactive	INACTIVE
(blue)	PCDI_MASTER non-redundant MODBUS TCP device is inactive	INACTIVE
(blue)	PCDI_MASTER non-redundant Safety Manager is inactive	INACTIVE
(blue)	PCDI_MASTER redundant bridges are inactive	INACTIVE/INACTIVE
(blue)	PCDI_MASTER redundant MODBUS TCP devices are inactive	INACTIVE/INACTIVE
(blue)	PCDI_MASTER redundant Safety Managers are inactive	INACTIVE/INACTIVE
(green)	PCDI_MASTER non-redundant bridge is active	OK
(green)	PCDI_MASTER non-redundant MODBUS TCP device is active	OK
(green)	PCDI_MASTER non-redundant Safety Manager is active	OK
(green)	PCDI_MASTER redundant bridges are active	OK/OK
(green)	PCDI_MASTER redundant MODBUS TCP devices are active	OK/OK
(green)	PCDI_MASTER redundant Safety Managers are active	OK/OK
(yellow)	PCDI_MASTER non-redundant bridge is not used	STBY

If Icon is . . .	Then, it represents . . .	And, Module State (IOMSTATE)* Is. . .
(yellow)	PCDI_MASTER non-redundant MODBUS TCP device is not used.	STBY
(yellow)	PCDI_MASTER non-redundant Safety Manager is not used	STBY
(yellow/yellow)	PCDI_MASTER redundant bridges are not used	STBY/STBY
(yellow/yellow)	PCDI_MASTER redundant MODBUS TCP devices are not used	STBY/STBY
(yellow/yellow)	PCDI_MASTER redundant Safety Managers are not used	STBY/STBY
(green/yellow)	PCDI_MASTER primary redundant bridge is active, secondary bridge is not used	OK/STBY
(green/yellow)	PCDI_MASTER primary redundant MODBUS TCP device is active, secondary device is not used	OK/STBY
(green/yellow)	PCDI_MASTER primary redundant Safety Manager is active, secondary Safety Manager is not used	OK/STBY
(yellow/green)	PCDI_MASTER primary redundant bridge is not used, secondary bridge is active	STBY/OK
(yellow/green)	PCDI_MASTER primary redundant MODBUS TCP device is not used, secondary device is active	STBY/OK
(green/yellow)	PCDI_MASTER primary redundant Safety Manager is not used, secondary Safety Manager is active	STBY/OK
(green/red/red exclamation)	PCDI_MASTER primary redundant bridge is active, secondary bridge has failed	OK/FAIL
(green/red/red exclamation)	PCDI_MASTER primary redundant MODBUS TCP device is active, secondary device has failed	OK/FAIL
(green/red/red exclamation)	PCDI_MASTER primary redundant Safety Manager is active, secondary Safety Manager has failed	OK/FAIL
(yellow/red/red exclamation)	PCDI_MASTER primary redundant bridge is not used, secondary bridge has failed	STBY/FAIL
(yellow/red/red exclamation)	PCDI_MASTER primary redundant MODBUS TCP device is not used, secondary device has failed	STBY/FAIL
(yellow/red/red exclamation)	PCDI_MASTER primary redundant Safety Manager is not used, secondary Safety Manager has failed	STBY/FAIL
(red/green/red exclamation)	PCDI_MASTER primary redundant bridge has failed, secondary bridge is active	FAIL/OK
(red/green/red exclamation)	PCDI_MASTER primary redundant MODBUS TCP device has failed, secondary device is active	FAIL/OK
(red/green/red exclamation)	PCDI_MASTER primary redundant Safety Manager has failed, secondary Safety Manager is active	FAIL/OK
(red/yellow/red exclamation)	PCDI_MASTER primary redundant bridge has failed, secondary bridge is not used	FAIL/STBY
(red/yellow/red exclamation)	PCDI_MASTER primary redundant MODBUS TCP device has failed, secondary device is not used	FAIL/STBY
(red/yellow/red exclamation)	PCDI_MASTER primary redundant Safety Manager has failed, secondary Safety Manager is not used	FAIL/STBY

If Icon is . . .	Then, it represents . . .	And, Module State (IOMSTATE)* Is. . .
(red/red/red exclamation)	PCDI_MASTER primary redundant bridge has failed, secondary bridge has failed	FAIL
(red/red/red exclamation)	PCDI_MASTER primary redundant MODBUS TCP device has failed, secondary device has failed	FAIL
(red/red/red exclamation)	PCDI_MASTER primary redundant Safety Manager has failed, secondary Safety Manager has failed	FAIL
(red/black exclamation)	PCDI_MASTER non-redundant bridge with communication failure	FAIL
(red/black exclamation)	PCDI_MASTER non-redundant MODBUS TCP device with communication failure	FAIL
(red/black exclamation)	PCDI_MASTER non-redundant Safety Manager with communication failure	FAIL
(red/red exclamation)	PCDI_MASTER non-redundant bridge has failed	FAIL
(red/red exclamation)	PCDI_MASTER non-redundant MODBUS TCP device has failed	FAIL
(red/red exclamation)	PCDI_MASTER non-redundant Safety Manager has failed	FAIL
* This parameter appears on the Module Statistics tab of the Module Properties form for the selected PCDI_MASTER block in Monitoring mode.		

6.2.3 Control Module block icon appearances

The following table summarizes the various appearances that a CM block icon in Control Builder can assume based on configuration, view, and current CM operating state and status.

If Icon is . . .	Then, it represents . . .	And, Execution State* Is. . .
Project Tab		
(gray)	CM configured for operation in Project.	N/A
Monitoring Tab		
(gray)	CM loaded, but not monitoring	N/A
(blue)	CM inactive	INACTIVE
(green)	CM Active	ACTIVE
(red/black exclamation)	CM with communication failure	N/A
* This appears in title bar on Module Properties form for selected Control Module in Monitoring mode.		

6.2.4 Monitoring/Interacting with given component/block

Once you download a Controller and its contents, you can use the **Monitoring** tab to interact with the components including the function blocks in the Control Module.

You simply double-click the desired component/block icon in the hierarchy tree menus under the Controller icon to call up the associated Configuration form or open the Control Module in the Control Drawing to double-click blocks contained in the CM. Click the given Configuration form tab to view the current status of the related block data. Both the PCDI_MASTER and CM function blocks must be active to view online data. The data can be viewed either by name or parameter reference by **not** checking or checking the **Show Parameter Names** check box at the bottom of each tab.

Please refer to the *Control Building Guide* sections in On-Line *Monitoring Using Control Builder* for detailed information:

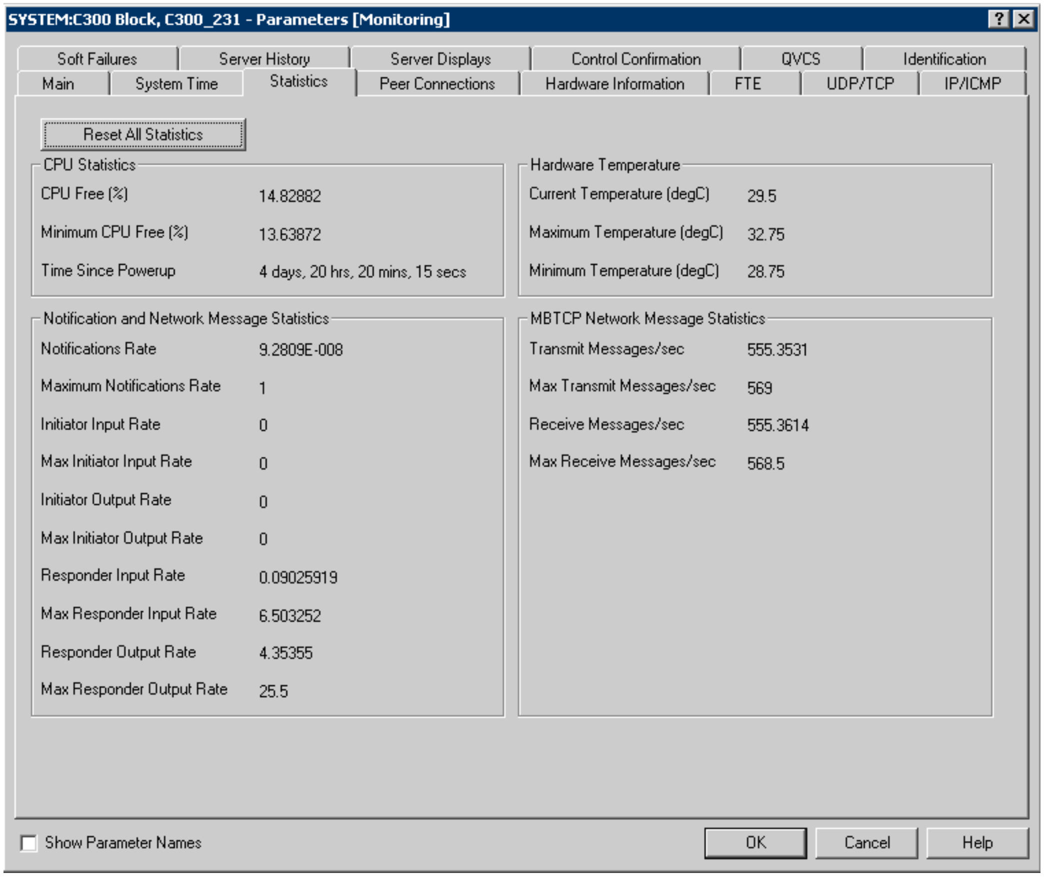
- *Setting the CEE Inactive*
- *Changing Parameters while Monitoring*

6.2.5 Monitoring PCDI related statistics through C300 block in Monitoring mode

Use the following steps to monitor performance counters for PCDI functions on the C300 block configuration form.

1 Double click the C300 block icon on the **Monitoring** tab to call up its configuration form.

2 Click the **Statistics** tab to display the tab.



SYSTEM:C300 Block, C300_231 - Parameters [Monitoring]

CPU Statistics

CPU Free (%)	14.82882
Minimum CPU Free (%)	13.63872
Time Since Powerup	4 days, 20 hrs, 20 mins, 15 secs

Hardware Temperature

Current Temperature (degC)	29.5
Maximum Temperature (degC)	32.75
Minimum Temperature (degC)	28.75

Notification and Network Message Statistics

Notifications Rate	9.2809E-008
Maximum Notifications Rate	1
Initiator Input Rate	0
Max Initiator Input Rate	0
Initiator Output Rate	0
Max Initiator Output Rate	0
Responder Input Rate	0.09025919
Max Responder Input Rate	6.503252
Responder Output Rate	4.35355
Max Responder Output Rate	25.5

MBTCP Network Message Statistics

Transmit Messages/sec	555.3531
Max Transmit Messages/sec	569
Receive Messages/sec	555.3614
Max Receive Messages/sec	568.5

☐ Show Parameter Names

3 Monitor the data in the *MBTCP Network Message Statistics* section for PCDI related functions performance.

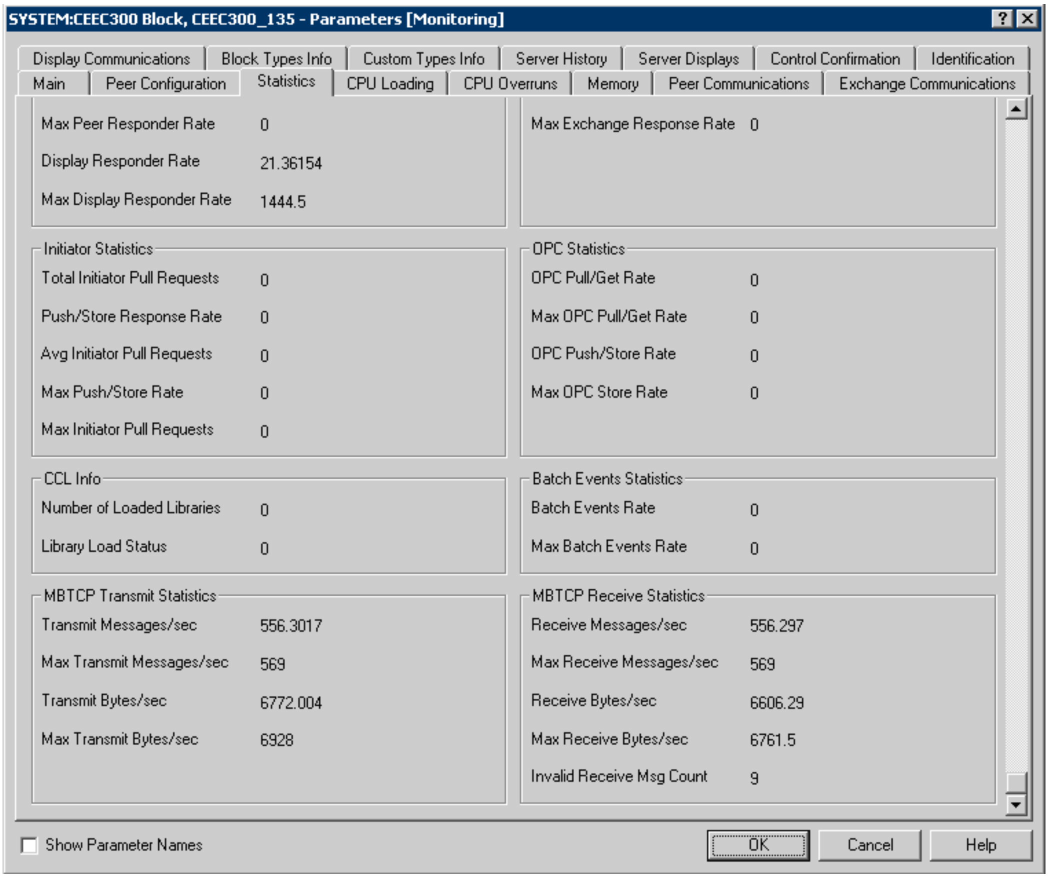
4 Click the **OK** button to close the form.

6.2.6 Monitoring PCDI related statistics through CEEC300 block in Monitoring mode

Use the following steps to monitor performance counters for PCDI functions on the CEEC300 block configuration form.

1 With C300 block tree hierarchy expanded, double click the CEEC300 block icon on the **Monitoring** tab to call up its configuration form.

2 Click the **Statistics** tab to display the tab and use the scroll bar to scroll to the bottom of the tab.



SYSTEM:CEEC300 Block, CEEC300_135 - Parameters [Monitoring]

Max Peer Responder Rate	0
Display Responder Rate	21.36154
Max Display Responder Rate	1444.5

Max Exchange Response Rate	0

Initiator Statistics

Total Initiator Pull Requests	0
Push/Store Response Rate	0
Avg Initiator Pull Requests	0
Max Push/Store Rate	0
Max Initiator Pull Requests	0

OPC Statistics

OPC Pull/Get Rate	0
Max OPC Pull/Get Rate	0
OPC Push/Store Rate	0
Max OPC Store Rate	0

CCL Info

Number of Loaded Libraries	0
Library Load Status	0

Batch Events Statistics

Batch Events Rate	0
Max Batch Events Rate	0

MBTCP Transmit Statistics

Transmit Messages/sec	556.3017
Max Transmit Messages/sec	569
Transmit Bytes/sec	6772.004
Max Transmit Bytes/sec	6928

MBTCP Receive Statistics

Receive Messages/sec	556.297
Max Receive Messages/sec	569
Receive Bytes/sec	6606.29
Max Receive Bytes/sec	6761.5
Invalid Receive Msg Count	9

☐ Show Parameter Names

3 Monitor the data in the *MBTCP Transmit Statistics* and *MBTCP Receive Statistics* sections for PCDI related functions performance.

4 Click the **OK** button to close the form.

6.3 Initiating Switchover of Redundant Devices

You can manually initiate a switchover of redundant peer devices through the **Monitoring** tab in Control Builder.

A peer device fail over occurs when the maximum number of requests for the device have been attempted. This can be caused when the device time out occurs on each of the retries or if the device returns a failure after each retry

If you have Engineer security level or above, you can change the Connection to Use (CONTOUSE) parameter configuration in **Monitoring** mode to force the connection in use and disable the connection not selected. All connection attempts and alarms will be disabled on the non-selected connection. This function may be useful for maintenance purposes.

Prerequisites

- You have logged on with sufficient privileges to make changes in a control strategy.
- You have loaded a control strategy to the Controller and can view the PCDI_MASTER device through the **Monitoring** tab of Control Builder.

To initiate manual switchover

1. Double-click the PCDI_MASTER device icon (green) to call up its configuration properties form.
2. Click the **Module Configuration** tab to display it.
3. Click the down-arrow in the **Preferred Connection to use** box and select **Secondary** from the list. Confirm the action.
4. A switch-over is attempted. If an error occurs, the connection switches back to the best connection available. The **Connection in Use** box displays the resulting connection.
5. This completes the procedure. Go to the next section.

6.4 Checking license details

Please refer to the *Control Hardware Planning Guide* for more information about viewing application licenses and features.

6.5 Response to C300 RAM retention restart

During a C300 RAM retention restart (RRR), all block request states are scanned and any non-pending or completed requests are set to error condition to force retransmission in the first block execution cycle. Request state will be available in battery backed PCI RAM. If the pending requests are not received (missed) due to excessive restart time or because they were sent while the node was unavailable, the requests are retransmitted and if they have reached the maximum retry count, they will timeout.

6.6 PCDI support for checkpoint save/restore functions

The Checkpoint function in Control Builder lets you save and restore operation and configuration data associated with a controller. The primary purpose of Checkpoint's save and restore functions are to provide the operator with the capability to bring a failed controller back to an operational condition as soon as possible. There are other ancillary scheduling, archiving and rebuilding functions provided by Checkpoint. Refer to the Checkpoint Reference section the *Control Building User's Guide* for more information about the checkpoint function.

While the operation and configuration data associated with PCDI supports the standard Experion checkpoint save/restore functions, note the following exceptions for the given PCDI parameter data.

- A checkpoint save operation will not save any run-time values for the following PCDI related parameters. It will only save the configuration values for these parameters.

PCDI Parameter	PCDI Parameter	PCDI Parameter
DEFTIMOUT	LOOPTYPE	UIDLOOPMODE
MAXPENDREQ	LOOPADDR	UIDLOOPTYPE
MAXUIDREQ	LOOPDATA	UIDLOOPADDR
REQRTRY	LOOPDATACHG	UIDLOOPDATA
MSGDELAY	PREFERREDCONN	UIDLOOPDATACHG
LOOPRATE	TIMOUT	
LOOPMODE	UIDLOOPRATE	

- A checkpoint save operation does not save a parameter that is not exposed during configuration even if a run-time configuration change exposes the parameter. For example, when LOOPTYPE is configured as *Diag Register*, both LOOPADDR and LOOPDATA are not available/exposed. If the LOOPTYPE configuration is changed to Coil during run-time, both LOOPADDR and LOOPDATA are available/exposed; but they will not be saved in the checkpoint file.

7 Peer Control Data Interface maintenance

Related topics

"Periodic checks" on page 102

7.1 Periodic checks

The following table identifies checks that you should make periodically (every 3 to 6 months) to keep the peer control data interface in good working condition.

Check . . .	Possible Corrective Action . . .
That all connections are secure.	Secure connections, as needed.
That cable insulation is not worn or cracked.	Replace cables, as required.
That third-party equipment is being maintained.	Follow vendor maintenance recommendations.

8 Peer Control Data Interface troubleshooting

Related topics

8.1 Isolating problems

This section offers some general data about fault classifications and initial checks that may help you isolate the cause of a problem.

If you have observed a specific symptom and/or identified an error code, go to the Fixing Common Problems section to see if you can find a matching topic.

8.2 Fault classifications

The following table lists some possible ways for classifying faults along with some identifying characteristics.

Fault Classification	Characteristics
Hard Failure	Failure detected by hardware; operation cannot continue. If the fault does not prevent software processing the problem, the affected node will be rebooted under software control into the FAIL State. • If failure occurs on Primary node, it triggers a switchover to synchronized Secondary node. • If failure occurs on Secondary node, it results in loss of synchronization and reduced availability until the problem is resolved. • If failure occurs on non-redundant node, it results in loss of control and loss of view.
Severe Failure	Failure detected by software; operation cannot continue. The affected node will be rebooted under software control into the FAIL State. The failure scenarios are as listed in the row above for the Hard Failure.
Soft Failure	Failure detected by software; operation continues with full control and full view. Soft failures are alarmed to the operator. FTE will be monitored by the FTE System Management Tool. • If failure occurs on Primary node, it does not trigger a switchover to synchronized Secondary node. • If failure occurs on Secondary node, it does not result in loss of synchronization. • If failure occurs on non-redundant node, it does not result in loss of control and loss of view.
Installation/Startup Failure	Detected by software. Node may not become operational. • This failure does not apply to synchronized Primary node, since installation and startup must be successful to reach synchronized Primary state. • If failure occurs on Secondary node, it results in the inability to complete the initial synchronization and to view the node on the network. • If failure occurs on non-redundant node, it results in the inability to commence control and to view the node on the network.
Communication	Communication errors between peer nodes and/or I/O devices, including Fault Tolerant Ethernet Bridge (FTEB), do not cause any node state change.

8.3 Initial checks

This section offers some checks that you can make to help isolate the problem. The checks are arranged in no particular order.

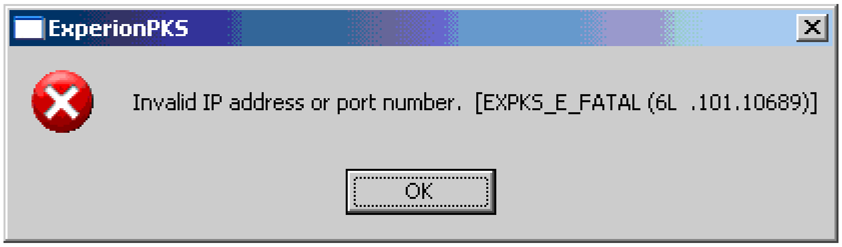
Related topics

“Checking Control Builder error code reference” on page 106
“Viewing release information log” on page 106
“Viewing trace log” on page 106
“Checking version and revision log” on page 106
“Checking server point build log” on page 107
“Checking server point build error log” on page 107
“Checking error log” on page 107

8.3.1 Checking Control Builder error code reference

An indication of a problem may be in the form of an error dialog that includes an error message and possibly an error code in Control Builder.

The syntax for a typical Control Builder error message is as shown in the following illustration:



Invalid IP address or port number. [EXPKS_E_FATAL (6L .101.10689)]

In this syntax, the error code is the last five digits in the message or **10689**.

Please refer to the *Control Builder Error Codes Reference* book for applicable error code information.

8.3.2 Viewing release information log

The ReleaseInfo.txt log provides a list of Experion software releases that have been installed on the computer.

To view the log, navigate to this file location on the server: C:\Program Files\Honeywell\Experion PKS \Engineering Tools\system\bin\ReleaseInfo.txt.

8.3.3 Viewing trace log

The TraceLogRs.txt log provides a list of definitions for strings associated with *breadcrumbs* data for given hardware components. The *breadcrumbs* provide a way to trace operations leading up to an event.

To view the log, navigate to this file location on the server: C:\Program Files\Honeywell\Experion PKS \Engineering Tools\system\bin\TraceLogRs.txt.

8.3.4 Checking version and revision log

The ver_rev.txt log provides a list of components by model number with software version/revision along with compatible Experion software release(s).

To check the log, navigate to this file location on the server: C:\Program Files\Honeywell\Experion PKS \Engineering Tools\system\bin\ver_rev.txt.

8.3.5 Checking server point build log

The SvrPtBld.txt log provides list of process (CB) points built in the server database.

To check the log, navigate to this file location on the server: C:\Program Files\Honeywell\Experion PKS \Engineering Tools\temp\SvrPtBld.txt.

8.3.6 Checking server point build error log

The svrptblderr.txt log provides list of any errors associated with process (CB) points built in the server database

To check the log, navigate to this file location on the server: C:\Program Files\Honeywell\Experion PKS \Engineering Tools\temp\svrptblderr.txt.

8.3.7 Checking error log

The Errlog_n.txt log provides a running list of Control Builder detected errors in chronological order. The n represents any number that is assigned to the most recent log.

To check the log, click **View** > **Error Log** in the Control Builder window to open the log.

8.4 Fixing common problems

This section identifies some common problems and describes how you might fix them.

Related topics

"Loss of power" on page 108

"Loss of communications" on page 108

8.4.1 Loss of power

The power supply has failed or the main power source has been shut down or is experiencing a *brownout* or *blackout* condition.

Diagnostic Check	In the **Monitoring** tab, the PCDI_MASTER device icon turns red.
Cause	Main power source has been disconnected or shut down either manually or temporarily by *brownout* or *blackout* condition.
Solution	Re-connect the main power source or turn it On or wait for temporary *brownout* or *blackout* condition to pass.

8.4.2 Loss of communications

No device data is being processed

Diagnostic Check	In the **Monitoring** tab, the PCDI_MASTER device icon turns red. In Station Event Summary, event error is logged
Cause 1	Broken or disconnected cable.
Solution	Repair broken cable or connect cable, as applicable.
Cause 2	Intermittent communications timeout in device.
Solution	For redundant devices, switchover to secondary device. Try cycling power to the device. Replace the device, if fault persists. Initiate a MODBUS loopback test on each end device to check if device can be reached.

8.5 Getting further assistance

Related topics

"Other troubleshooting sources" on page 109

"Guidelines for requesting support" on page 110

8.5.1 Other troubleshooting sources

The following table lists other documents and sections that contain troubleshooting information for other Experion subsystems.

Document/Section	Comments
Experion Rxxx > Reference	There is a separate interface reference for each type of controller other than the Process Controller; for example, the *ASEA Interface Reference.* Most of these references contain an interface-specific troubleshooting section.
Experion Rxxx > Reference > Control Builder Error Codes Reference	Describes error codes generated from within Control Builder.
Experion Rxxx > Troubleshooting and Maintenance > Control Hardware Troubleshooting and Maintenance Guide	The main repository for troubleshooting, maintenance and repair of Process Controllers.
Experion Rxxx > Configuration > DeviceNet Implementation Guide > Troubleshooting DeviceNet Status Failures	Describes error codes generated from DeviceNet Interface Board.
Experion Rxxx > Installation and Upgrades > Fault Tolerant Ethernet Bridge Implementation Guide > Service > Troubleshooting	Troubleshooting FTE bridges.
Experion Rxxx > Installation and Upgrades > Fault Tolerant Ethernet Installation and Service Guide > Troubleshooting FTE Nodes	Troubleshooting FTE nodes.
Experion Rxxx> Reference > Honeywell TDC 3000 Data Hiway Interface Reference >TDC error codes and Troubleshooting	Troubleshooting TDC 3000 Hiway problems.
Experion Rxxx > Configuration > Qualification and Version Control System User Guide > QVCS Troubleshooting	Troubleshooting QVCS.
Experion Rxxx > Operations > SafeView User's Guide > Appendix D - SafeView Error Messages	Describes the meaning of SafeView configuration errors.
Experion Rxxx > Reference > Server Scripting Reference > Server scripting error messages	Describes the meaning of error messages in the server log specific to server scripting.
Experion Rxxx > Reference > System Management Configuration Guide > Troubleshooting System Management	Describes the meaning of System Management Configuration errors.
Experion Rxxx > Reference > System Management Configuration Guide > Troubleshooting SES	Describes the meaning of SES Configuration errors.
Experion Rxxx > Reference > System Management Configuration Guide > Troubleshooting SPS	Describes the meaning of SPS Configuration errors.
Experion Rxxx > Planning and Design > Planning, Installation, and Service for WS360	Troubleshooting workstation nodes used in Experion and TPN.

8.5.2 Guidelines for requesting support

If you cannot resolve a problem using this guide, you can request support from your Honeywell Solution Support Center.

When requesting support, please supply as many relevant details as possible, including:

- **Short summary of the problem**
- **Product Name and release.**
- **Recent changes**, such as upgrades/service packs, to Experion software, Windows or other applications.
- **Subsystem and its version/build**, if the problem relates to a particular subsystem, such as Station or Quick Builder.

  If the problem relates to **Display Builder**, please specify whether it is HMIWeb Display Builder (for HMIWeb displays) or Display Builder (for DSP displays).
- **Operating system, variant and service pack,** for example 'Windows 2000 Server, SP5'.
- **Instructions on how to reproduce the problem.** If the problem is reproducible, please supply step-by-step instructions - the more detailed the steps the better.
- **Diagnostic package which contains any relevant logs.**

9 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.

9.1 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.

9.2 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.

9.3 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.

9.4 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.