

Experion Controller References Volume 2 of 3

EP-DSX114

300

11/05

Release 300

Document	Release	Issue	Date
EP-DSX114	300	0	November 2005

Notice

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 Limited Australia.

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.

Copyright 2005 – Honeywell Limited Australia

Honeywell trademarks

PlantScape[®], SafeBrowse[®], **TotalPlant**[®] and TDC 3000[®] are U.S. registered trademarks of Honeywell International Inc.

Experion[™] 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.

Support and other contacts

United States and Canada

Contact Honeywell IAC Solution Support Center
Phone 1-800 822-7673. In Arizona: (602) 313-5558
Calls are answered by dispatcher between 6:00 am and 4:00 pm Mountain Standard Time. Emergency calls outside normal working hours are received by an answering service and returned within one hour.
Facsimile (602) 313-5476
Mail Honeywell IS TAC, MS P13
2500 West Union Hills Drive
Phoenix, AZ, 85027

Europe

Contact Honeywell TAC-EMEA
Phone +32-2-728-2704
Facsimile +32-2-728-2696
Mail Honeywell TAC-EMEA
Avenue du Bourget, 1
B-1140 Brussels, Belgium

Pacific

Contact Honeywell Global TAC - Pacific
Phone 1300-300-4822 (toll free within Australia)
+61-8-9362-9559 (outside Australia)
Facsimile +61-8-9362-9169
Mail Honeywell Global TAC - Pacific
5 Kitchener Way
Burswood, WA, 6100, Australia
Email GTAC@honeywell.com

India

Contact Honeywell Global TAC - India
Phone +91-20-2682-2458 / 1600-44-5152
Facsimile +91-20-2687-8369
Mail Honeywell Automation India Ltd.
56 & 57, Hadapsar Industrial Estate
Hadapsar, Pune -411 013, India
Email Global-TAC-India@honeywell.com

Korea

Contact Honeywell Global TAC - Korea
Phone +82-2-799-6317
Facsimile +82-2-792-9015
Mail Honeywell Korea,
17F, Kikje Center B/D,
191, Hangangro-2Ga
Yongsan-gu, Seoul, 140-702, Korea
Email Global-TAC-Korea@honeywell.com

People's Republic of China

Contact Honeywell Global TAC - China
Phone +86-10-8458-3280 ext. 361
Mail Honeywell Tianjin Limited
17 B/F Eagle Plaza
26 Xiaoyhun Road
Chaoyang District
Beijing 100016, People's Republic of China
Email Global-TAC-China@honeywell.com

Singapore

Contact Honeywell Global TAC - South East Asia
Phone +65-6580-3500
Facsimile +65-6580-3501
+65-6445-3033
Mail Honeywell Private Limited
Honeywell Building
17, Changi Business Park Central 1
Singapore 486073
Email GTAC-SEA@honeywell.com

Taiwan

Contact Honeywell Global TAC - Taiwan
Phone +886-7-323-5900
Facsimile +886-7-323-5895
+886-7-322-6915
Mail Honeywell Taiwan Ltd.
10F-2/366, Po Ai First Rd.
Kaohsiung, Taiwan, ROC
Email Global-TAC-Taiwan@honeywell.com

Japan

Contact Honeywell Global TAC - Japan
Phone +81-3-5440-1303
Facsimile +81-3-5440-1430
Mail Honeywell K.K.
1-14-6 Shibaura Minato-Ku
Tokyo 105-0023
Japan
Email Global-TAC-JapanJA25@honeywell.com

Elsewhere

Call your nearest Honeywell office.

World Wide Web

To access Honeywell Solution Support Online, do the following:

- 1 In your web browser, type the address <http://www.honeywell.com/ps>.

- 2 Click **Login to My Account** and then log on.
- 3 Move the pointer over **Contacts & Support** in the top menu bar and then choose **Support** from the popup menu.

Contents

The Controller Reference Volumes 1, 2, and 3 contain separate “reference books” for each of the following controllers. Each book is entirely self-contained, with its own table of contents and index.

Volume 1	Volume 2	Volume 3
Allen-Bradley Interface Reference	Honeywell FSC Integration Reference	Honeywell UDC Integration Reference
Applicom Interface Reference	Honeywell LCS620 Interface Reference	Honeywell Universal Modbus Interface Reference
ASEA interface Reference	Honeywell Micromax Interface Reference	Modbus Interface Reference
Bristol Babcock Interface Reference	Honeywell RM7800 Flame Safeguard Interface Reference	Moore APACS Interface Reference
Bristol Babcock OpenBSI Interface Reference	Honeywell Series 9000 Integration Reference	Moore Mycro Interface Reference
DNP3 Protocol Interface Reference	Honeywell Safety Manager Integration Reference	OPC Client Interface
GE Fanuc Series 90 Interface Reference	Honeywell TDC 3000 Data Hiway Interface Reference	Siemens S5 and S7 Interface Reference
GEC GEM80 Interface Reference		Yamatake MA500 Interface Reference
Hitachi Interface Reference		
Honeywell DPR Interface Reference		
Honeywell EXCEL 5000 Interface Reference		

Honeywell FSC Integration Reference

Issue

5.1

Date

November 2005

Notice

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 Limited Australia.

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.

Copyright 2005 – Honeywell Limited Australia

Honeywell trademarks

PlantScape[®], SafeBrowse[®], **TotalPlant**[®] and TDC 3000[®] are U.S. registered trademarks of Honeywell International Inc.

Experion[™] and Honeywell Enterprise Buildings Integrator[™] are trademarks 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.

Support and other contacts

For technical assistance, call your nearest Honeywell office.

Contents

1	Getting started	5
	Support and documentation for FSC	6
	Serial FSC models supported	6
	Ethernet FSC models supported	6
	Documentation for FSC	6
	FSC-specific terms	7
2	FSC controller setup	9
	Differences between Serial and Ethernet controllers	10
	Supported architectures	11
	Serial FSC controllers	11
	Ethernet FSC controllers	13
	Communication settings	16
	Serial settings	16
	Ethernet settings	17
	Assigning SER numbers for sequence of events	19
	Reserved SER numbers	19
	Interpreting SOE lines	20
3	FSC controller configuration and addressing	23
	Defining a FSC channel	24
	FSC channel Main properties	24
	FSC Port properties	25
	Defining an FSC controller	29
	FSC controller Main properties	29
	Defining an FSC address for a point parameter value	32
	Entering an address	32
	Optimizing scanning performance	37
4	Server and Station tasks for FSC	39
	Accessing diagnostics for an Ethernet controller	40
	System information	40
	Extended diagnostics	40
	Testing communications with the server	41
	Testing communications	41
	Troubleshooting FSC communication errors for a serial controller	42
	Troubleshooting FSC communication errors for an Ethernet controller	42

Getting started

1

This reference describes how to set up, configure, and test FSC controller communications with the server. There is detailed information for defining the controller using Quick Builder.

Complete each step before commencing the next.

Steps for connecting and configuring an FSC controller.

Steps	Go to:
Determine FSC configuration	page 11
Setting up the communications parameters using the FSC configuration software	page 16
Use FSC configuration software to set SER numbers	page 19
Define channels with Quick Builder	page 24
Define controllers with Quick Builder	page 29
Download channel and controller definitions to the server	
Test communications	page 41
Troubleshooting communication errors	page 42
Define points with Quick Builder	page 32

Support and documentation for FSC

The server supports serial and Ethernet FSC controllers.

The server communicates with:

- Serial controllers using a point-to-point or multi-drop RS-232 link and FSC Modbus protocol
- Ethernet controllers using a proprietary protocol based on the Modbus protocol

Serial FSC models supported

The server supports the following serial FSC controller configurations:

FSC-100	A single rack.
FSC-100R	Two identical racks. Basically parallel FSC-100 configuration.
FSC-101	May consist of several racks.
FSC-102	Two Central parts and single I/O.
FSC-101R	Two Central parts and redundant I/O.
FSC-202	Two Central racks and redundant I/O.

Ethernet FSC models supported

The server supports the following Ethernet FSC controller configurations:

FSC-1oo1D	Single Central Part with single Ethernet connection
FSC-1oo1D	Single Central Part with redundant Ethernet connection
FSC-1oo2D	Redundant Central Parts with single Ethernet connection to each Central Part

Documentation for FSC

This reference only provides supplemental information, to interface FSC controllers with the server. For detailed information about installing and configuring FSC controllers, see the manufacturer's documentation.

FSC-specific terms

FSC

Fail Safe Controller.

SER channel

An FSC setting required so that the server can extract SOE data from the controller. The server uses the SER channel to poll for SOE data.

SOE

Sequence of Events.

Base SER Number

The unique ID assigned to a serial (non-Ethernet) controller so that the server can extract SOE data from that controller.

FSC controller setup

2

This chapter describes how to set up an FSC controller for the system.

The tasks for setting up an FSC controller are:

For	Go to:
Differences between serial and Ethernet controllers	page 10
Supported architectures	page 11
Communication settings	page 16
Using FSC configuration software to assign SER numbers	page 19

Differences between Serial and Ethernet controllers

The server supports serial and Ethernet FSC controllers. However, these two controller types are different, and following differences apply:

- Serial controllers use Modbus; whereas Ethernet controllers use a proprietary protocol based on Modbus.
- The diagnostic scan rate is not configurable in Ethernet controllers.
- The Base SER is not required for Ethernet controllers.
- There is a separate test utility for each type of controller.
- System Information and Extended Diagnostics are available for Ethernet controllers.

Supported architectures

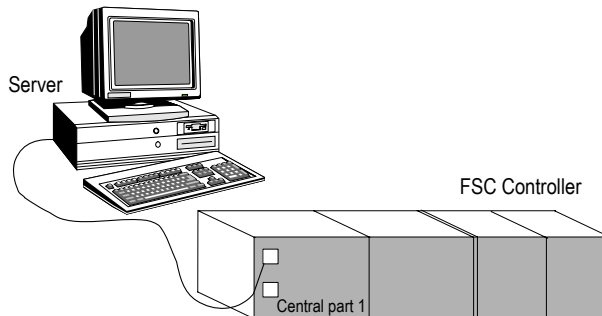
Serial FSC controllers

The server communicates with serial controllers using a point-to-point or multi-drop RS-232 link and FSC Modbus protocol.

Point-to-point configuration

A point-to-point configuration consists of a controller with a single COM module connected to a port on the server through a serial link.

Figure 1 Point-to-point configuration

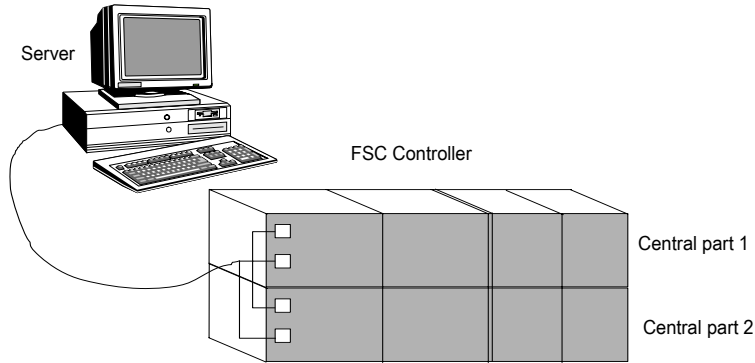


Multi-drop configuration

A multi-drop configuration consists of a redundant FSC controller with two COM modules connected on the same serial link which connects to a single port on the server.

Note that this type of configuration does not constitute server communications redundancy, but it does provide a means of talking to a redundant FSC system across a single channel.

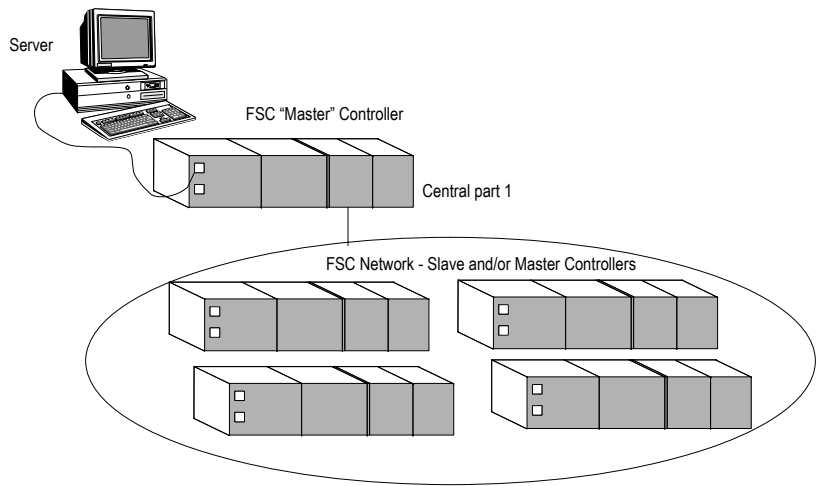
Figure 2 Multi-drop configuration



FSC network configuration

Using an FSC network configuration, a single FSC controller manages the data that the server requests or controls. One controller, designated as a “master” controller, is responsible for gathering SOE data from the FSC network. This enables SOE data from any networked controller to be obtained from the master controller.

A master controller is also useful for time synchronization. The master controller is responsible for time synchronization on the FSC network—synchronizing the time on the master controller also synchronizes the time on “slave” controllers. See “Time synchronization” on page 31.

Figure 3 FSC Network

Redundancy for serial controllers

Serial FSC controllers support communication redundancy. This is not the same as the multi-drop configuration. Communication redundancy involves a separate physical connection to communicate with the FSC controller.

Ethernet FSC controllers

The server communicates with Ethernet controllers using a proprietary protocol based on the Modbus protocol.



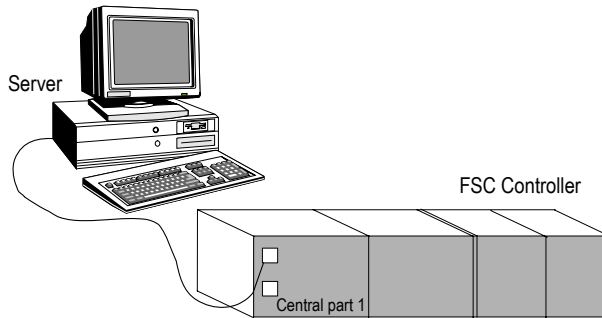
Note

Connecting or configuring a single FSC Ethernet controller to multiple servers is not a supported configuration.

Point-to-point configuration

A point-to-point configuration consists of a controller with a single COM module connected to a port on the server through an Ethernet link.

Figure 4 Point to point configuration

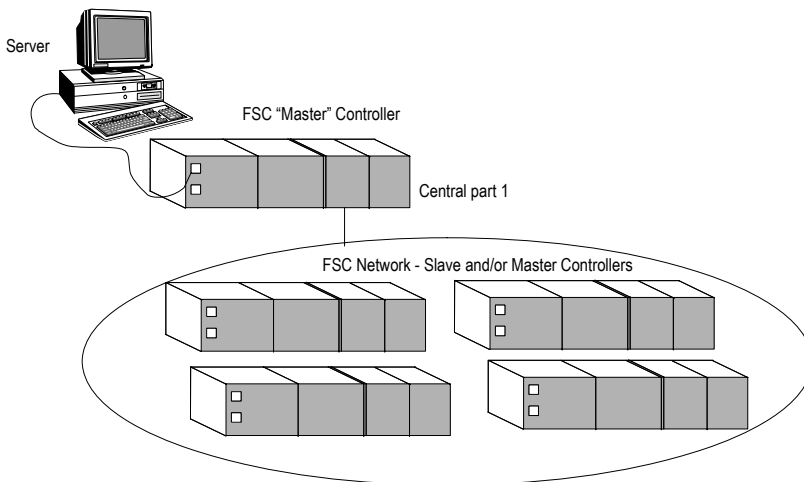


FSC network configuration

Using an FSC network configuration, a single FSC controller manages the data that the server requests or controls. One controller, designated as a “master” controller, is responsible for gathering SOE data from the FSC network. This enables SOE data from any networked controller to be obtained from the master controller.

A master controller is also useful for time synchronization. The master controller is responsible for time synchronization on the FSC network—synchronizing the time on the master controller also synchronizes the time on “slave” controllers. See “Time synchronization” on page 31.

Figure 5 FSC network



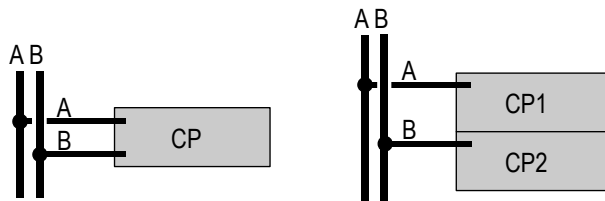
Redundancy for Ethernet controllers

Ethernet FSC controllers support communication redundancy. Communication redundancy involves a separate physical connection to communicate with the FSC controller.

The server supports the following redundancy options:

- Redundant connection to a single Central Part via a single, redundant 10018/E/E COM module
- Redundant connection to redundant Central Parts via a single, non-redundant 10018/E/1. COM module on each Central Part

Figure 6 Redundancy options



Communication settings

The communication settings are specific to the controller type (Serial or Ethernet).

Serial settings

The RS-232 connection is made to the controller's COM module, which can be of the following types:

- 10004/./.
- 10014/./.

Communication address

To communicate with a serial FSC controller, the server uses a Communication address, which can be calculated using the FSC System number specified when configuring the controller.

You use the FSC configuration utility, FSC Navigator, to specify the FSC System number, which must be unique within a system of FSC controllers.

The Communication address is equal to 4 times the FSC System number.

Modbus data tables and function codes

Serial FSC controllers use the following Modbus data tables and function codes.

In Quick Builder, you need to define a separate “logical” controller for each data table to which the server needs access. For example, if the server needs access to the Discrete Output and Input Register tables, you need to define two logical controllers.

Table 1 Modbus data tables

Data table	Server addressable range
Discrete (digital) Output (coils)	00001 to 08192 (read/write)
Discrete (digital) Input (contacts)	10001 to 18192 (read only)
Input Register	30001 to 38192 (read only)
Holding Register	40001 to 48192 (read/write)

Serial FSC controllers support the following Modbus protocol function codes:

Table 2 Modbus function codes

Function Code	Description
01	Read output status
02	Read input status
03	Read output registers
04	Read input registers
05	Force Single Coil
06	Preset Single Register

Ethernet settings

The Ethernet connection is made to the controller's 10018/E/1 or 10018/E/E COM modules.

Communication address

To communicate with an Ethernet FSC controller, the server uses a Communication address, which can be calculated using the FSC System number specified when configuring the controller.

You use the FSC configuration utility, FSC Navigator, to specify the FSC System number, which must be unique within a system of FSC controllers.

The Communication address is equal to 4 times the FSC System number.

Ethernet data tables

Ethernet FSC controllers use the following Modbus data tables.

In Quick Builder, you need to define a separate “logical” controller for each data table to which the server needs access. For example, if the server needs access to the Coils and Registers, you need to define two logical controllers.



Note

The following points can only be built on a SOE only controller:

- FSC variables with a SER sequence number, but without a DCS address. This includes FSC variables with a SER sequence number which are located in a FSC network. See “FSC network” on page 14.
- FSC system events. See “Reserved SER numbers” on page 19.

Table 3 Ethernet data tables

Data Table	Server Addressable Range (DCS address)
Coils (FSC types: I, O)	1 - 8192
SOE Only	N/A
Registers (FSC types: BI, AI, BO, AO)	10001 - 18192

Assigning SER numbers for sequence of events

FSC controllers have SOE (Sequence Of Events) capabilities. If you want the server to record a controller's SOEs in its SOE log, you need to assign an SER number to each internal address.



Note

In the case of a serial FSC controller, you need to define a Base SER for the controller.

To record a controller's SOEs in its SOE log:

- 1 Assign an SER number to each internal address you want the server to record. (It is recommended that you assign SER numbers in a tight block for each controller.)
You use the FSC configuration utility, FSC Navigator, to assign SER numbers.
- 2 When defining the controller in Quick Builder, select the **SOE Enable** check box. (If you define several "logical" controllers, only select this for one controller.) See "Defining an FSC controller" on page 29.
- 3 Assign a separate point, in Quick Builder, to each SER number. A point can only have 1 SER number associated with it.

Reserved SER numbers

FSC controllers reserve the following SER numbers for system events. If you want the server to log these system events, you need to assign points to them that are reserved for system events.

SER Number (serial)	SER Number (Ethernet)	System Event
Base SER	0	All forces cleared
Base SER + 1	1	FSC System Fault
Base SER + 2	2	SER buffer full
Base SER + 3	3	SER buffer Empty
Base SER + 5	5	Force Event

**Note**

- Base SER + 4 (4 in the case of Ethernet) is not a valid event.
- The “SER Buffer Empty” event cannot be stored in server SOE. If a point is defined with this SER number, it is ignored.
- To be logged on to the SOE display when using an Ethernet FSC controller, the System Events must be built on a SOE only controller. This is because System Events have no DCS address associated with them.

Interpreting SOE lines

The following fields of the server’s SOE file are populated for each SOE with a matching SER number:

Field	Description
Date & Time	The date and time given to the SOE by the controller.
Source	The server’s point ID that corresponds with this SOE.
Condition	The SER number associated with this SOE.
Action	Only applicable to Ethernet controllers. Information indicator dependent on the types of SER. See “Action descriptions” on page 20.
Description	The server’s point description.
Value	The value returned in the SOE (if applicable). For a status point built against a coil, this displays the appropriate state descriptor for that point. Indeterminate values display either blank or zero.

Action descriptions

Table 4 Action descriptions

Event Type	FSC Data Type	FSC Description	Server-action Field
Event Report ⁱ	Any		N/A ⁱⁱ
Process Variable Event	Boolean	True to False	F
		False to True	T
	Analog	Low	LO
		Healthy	HLT
		High	HI

Table 4 Action descriptions (*continued*)

Event Type	FSC Data Type	FSC Description	Server-action Field
Process Variable Force Event	Boolean	Force Set	SET
		Force Cleared	CLR
	Analog	Force Set	SET
		Force Cleared	CLR
Diagnostic Event	Diagnostic Event		N/A

- i System events. See “Reserved SER numbers” on page 19.
- ii Actions that are not applicable or indeterminate are displayed as “--”.

FSC controller configuration and addressing

3

This chapter describes how to configure an FSC controller for your system using Quick Builder.

The Quick Builder controller configuration tasks are described:

For:	Go to:
Defining an FSC channel	page 24
Defining an FSC controller	page 29
Defining an address for a point parameter value	page 32
Creating points in Quick Builder after downloading channels and controllers	page 32
Optimizing scanning performance	page 37

Defining a FSC channel



Note

If you use both serial and Ethernet FSC controllers, you need to create a separate channel for each type of controller.

To define a channel using Quick Builder:

- 1 Click  to add a channel to display the Add Items dialog box.
- 2 Select `channel` as the item and `FSC` as the channel type.
- 3 Enter the channel details on the Main property page for the channel. See “FSC channel Main properties” on page 24.
- 4 Click the Port tab, and select a **Port Type** appropriate to the controller type. Define the other properties as required. See “FSC Port properties” on page 25.

FSC channel Main properties

Use the Main tab to enter the basic channel properties for an FSC channel.

Property	Description
Name	A unique name for the channel. A maximum of 10 alphanumeric characters.(Double quotation marks and spaces are not allowed.)
Description	(Optional) A description of the channel. A maximum of 30 characters can be used, including spaces.
Marginal Alarm Limit	The communications alarm marginal limit at which the channel is declared to be marginal. When this limit is reached, a high priority alarm is generated. A channel barometer monitors the total number of requests and the number of times the controller did not respond or response was incorrect. The barometer increments by 2 or more, depending on the error and decrements for each good call. To calculate an acceptable limit, multiply the square root of the number of controllers on the channel by the Marginal Alarm Limit defined for those controllers. (Normally, you specify the same value for all controllers on a channel). For example, if there are 9 controllers on the channel and their Marginal Alarm Limit is set to 10, the value would be $[3 \text{ is square root of } 9] \times 10 = 30$.

Property	Description
Fail Alarm Limit	The communications alarm fail limit at which the channel is declared to have failed. When this barometer limit is reached, an urgent alarm is generated. Set this to double the value specified for the channel Marginal Alarm Limit.
Connect Timeout	The time (in seconds) the server waits to connect to the controller before abandoning the connection. The default is 10 seconds. Use the default value unless the communications line has a high error rate or unless you are using modems.
Read Timeout	The time (in seconds) that the server waits for a reply from the controller. The default is 2 seconds. Use the default value unless the communications line has a high error rate or unless you are using modems.
Item Type	Shows the type of item specified when this item was created.
Last Modified	Shows the date of the most recent modification to this channel's property details.
Item Number	The unique item number currently assigned to this item by Quick Builder. You can change the item number if you need to match your current server database configuration. The item number must be between 1 and the maximum number of channels allowed for your system.

FSC Port properties

The Port tab defines the communication-related properties for a channel. The **Port Type** for FSC controllers can be:

- **serial**. Only applicable to a directly-connected serial FSC controller. A serial communications interface, such as RS-232. See “Serial port properties” on page 26.
- **TerminalServer**. Only applicable to a serial FSC controller that communicates with the server via a terminal server. See “Terminal server port properties” on page 27.
- **LANVendor**. Only applicable to an Ethernet FSC controller. See “LANVendor port properties” on page 28.

Serial port properties



Note

Only applicable to serial FSC controllers.

Property	Description
Serial Port Name	The device name of the serial port.
Baud Rate	The number of data bits per second. The default is 9600.
Number of Data Bits	The number of data bits used for transmission. The default is 8.
Stop Bits	The number of stop bits used for transmission. The default is 1.
Parity	The parity verification used on the port. The default is NONE.
Checksum	The type of checksum error detection used for the port. Select the value that matches the setting on the communication device: • CRC16_0 or CRC16_1 (if Cyclic Redundancy Check (CRC) is set) • ONESCOMP or TWOSCOMP (if Longitudinal Redundancy Check (LRC) is set) • XOR (If exclusive or is set)
XON/XOFF	The type of XON/XOFF software flow control used to stop a receiver from being overrun with messages from a sender. The types are: • None (default) • Input (use XON/XOFF to control the flow of data on the receive line) • Output (use XON/XOFF to control the flow of data on the transmit line)

Property	Description
Handshaking Options	For RS-232 • Enable RTS/CTS flow control. Stops a receiver from being overrun with messages from a sender by using RTS/CTS flow control. • Detect DCD. Select if the Data Carrier Detect communication status line of the COM port requires monitoring (usually when using modem or microwave linking). When selected, the communications fails if the desired COM status line is not high—for example, on a dial-up link connection for a modem. • Detect DSR. Select if the Data Set Ready communication status line of the COM port requires monitoring (usually when using modem or microwave linking). When selected, the communications fails if the desired COM status is not achieved. Note: No options available for RS-422. For RS-485. The server does not support RS-485 for FSC controllers.

Terminal server port properties



Note

Only applicable to serial FSC controllers.

Property	Description
Terminal Server TCP Host Name	The name and port number of terminal server to which the channel is connected.
Terminal Server TCP Port No	You can specify either a TCP host name or an IP address, but it must match the TCP host name used when you installed and internally configured the terminal server.
Idle Timeout	The time, in seconds, the channel waits for a successful connection to the server before closing the connection. A value of 0 indicates that the connection is never closed.

Property	Description
Checksum	The type of checksum error detection used for the port. Select the value that matches the setting on the communication device: • CRC16_0 or CRC16_1 (if Cyclic Redundancy Check (CRC) is set) • ONESCOMP or TWOSCOMP (if Longitudinal Redundancy Check (LRC) is set) • XOR (If exclusive or is set)

LANVendor port properties



Note

Only applicable to Ethernet FSC controllers.

Property	Description
Port Name	Leave blank.

Redundant port properties

A communications link being used as a redundant link requires an additional port definition for the redundant port. After you complete the port definition, enter the same kind of port definition for the redundant port.

Defining an FSC controller

You need to define a controller to describe each PLC on a channel.

To define a controller using Quick Builder:

- 1 Click  to add a controller to display the Add Items dialog box.
- 2 Select `Controller` as the item and `FSC` as the controller type.
- 3 Select the appropriate channel from the **Channel Name** list. (When you select the channel, the appropriate serial or Ethernet properties appear.)
- 4 Define the controller's other properties. See "FSC controller Main properties" on page 29.

FSC controller Main properties

Use the Main tab to define the basic properties for an FSC controller.

Property	Description
Name	The unique name for the controller. A maximum of 10 alphanumeric characters. (Double quotation marks and spaces are not allowed.)
Description	A description for this controller. A maximum of 30 characters, including spaces.
Channel Name	The channel on which the controller communicates with the server. If you use both serial and Ethernet controllers, you must select a serial/terminal server channel for a serial controller, and an LAN Vendor channel for an Ethernet controller.
Marginal Alarm Limit	The communications alarm marginal limit at which the controller is declared to be marginal. When this value is reached, a high priority alarm is generated. This limit applies to the controller barometer which monitors the total number of requests to the controller and the number of times the controller did not respond or response was incorrect. The barometer increments by 2 or more, depending on the error and decrements for each good call. Type a new value in this field if you do not want the default, 25.

Property	Description
Fail Alarm Limit	The communications alarm fail limit at which the controller is declared to have failed. When this value is reached, an urgent alarm is generated. Set this to double the value specified for the controller Marginal Alarm Limit.
Communication Address	Set this to 4 times the FSC System number. The FSC System number is set using the FSC configuration utility, FSC Navigator. See “Communication address” on page 16.
FSC Type	Shows the controller type, which corresponds to the type of channel (serial or Ethernet) selected in Channel Name.
Data Table	The data table that this controller addresses. See: • “Modbus data tables and function codes” on page 16 for a serial controller • “Ethernet data tables” on page 17 for an Ethernet controller
Diagnostic	Only applicable to a serial controller. The value, in seconds, between diagnostic polls. The default is 60 seconds. To disable diagnostic polls, set the value to 0.
Base SER	Only applicable to a serial controller. The base address for this controller. See “Assigning SER numbers for sequence of events” on page 19.
IP Address 1 IP Address 2	The controller’s Ethernet addresses.
SOE Enable SOE Rate	Select SOE Enable if you want SOE data. Specify the time, in seconds, between polls for SOE data in SOE Rate. Only one logical controller per physical FSC controller can have SOE Enabled. If an SOE only controller is built, this should be the only logical controller with SOE enabled for that particular physical FSC controller. The SOE Rate must be a minimum of 5 seconds and defaults to 30 seconds. See “Assigning SER numbers for sequence of events” on page 19.

Property	Description
Synch Enable Sync Time	Select Synch Enable if you want to synchronize the controller time with the server time. Specify the time, in minutes, since midnight before synchronizing in Synch Time . The default is -1 (no synchronization). See “Time synchronization” on page 31.
Item Type	Shows the type of item specified when this item was created.
Last Modified	Shows the date of the most recent modification to this channel’s property details.
Item Number	The unique item number currently assigned to this item by Quick Builder. You can change the item number if you need to match your current server database configuration. The number must be between 1 and the maximum number of controllers allowed for your system.

Time synchronization

You can synchronize the time of FSC controllers with the server.

The **Synch Time** parameter is specified in minutes since midnight. When the assigned time is reached, the server sends a synchronization command with the current time to the controller.

If you define several “logical” controllers for a particular controller, you can specify a different synchronization time for each logical controller, in order to synchronize the physical controller at different intervals in the day.

Defining an FSC address for a point parameter value

Entering an address

The format for an FSC data table address is:

ControllerName Location

Part	Description
<i>ControllerName</i>	The name of the FSC controller.
<i>Location</i>	The location in the controller where the value is recorded. See “Location syntax for coils and registers” on page 32 for details.

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

Location syntax for coils and registers

Address [Format|BitNumber] [SER SerNumber]

Part	Description
<i>Address</i>	The server addressable memory address within the controller’s data table. For details, see: • “Modbus data tables and function codes” on page 16 (serial controller) • “Ethernet data tables” on page 17 (Ethernet controller)

Part	Description
<i>Format</i>	(Optional) Select an appropriate format to read raw values. To scale See “Scaling with data formats” on page 35. To read without scaling Select a format of C16, or enter a 0, to read all 16 bits in the register without scaling. To read less than 16 bits without scaling enter the starting bit number (1 to 15). If you are not using scaling, the point range is still used for PV indicator bar height only (the PV indicator bar is on the Point Detail display on Station). To read floating point values The supported floating point formats are specific to the controller type: - Serial, see “IEEEFP formats for serial controllers” on page 34. - Ethernet, see “Data formats supported for Ethernet controllers” on page 34. Select format IEEEFP to read two consecutive registers as a single precision floating point number.
<i>BitNumber</i>	(Optional) For Input register and Holding register data tables, a starting bit number can be specified. The valid range is 0 (default) to 15 where 0 is the right most bit in the 16-bit register. Analog and accumulator point parameters can read up to 16 bits. Status point parameters read 1, 2, or 3 consecutive bits.
<i>SerNumber</i>	The SER number. See “Assigning SER numbers for sequence of events” on page 19.



Note

For a status point with an OP configured with four output states (that is, across two consecutive coils), control of the OP is carried out by executing separate writes to each of the two coils. Therefore, a value is written to the first coil then another immediately to the second coil. FSC controller logic should be programmed to make allowance for these two separate writes operations.

Example

Analog point

PV source 161 U4095 SER 1000
 SP destination 162 U4095
 Mode destination 25 1

Status point

PV source 26

Accumulator point

PV source 171 C16

IEEEFP formats for serial controllers

Format	Description
IEEEFPB	Bytes are big endian format (this is the same as IEEEFP)
IEEEFPBB	Bytes are byte-swapped big endian format
IEEEFPPL	Bytes are little endian format
IEEEFPPLB	Bytes are byte-swapped little endian format



Note

- IEEEFP numbers use two data addresses to hold the number, address and address -1. Do not specify address -1 (the lower address) as the point parameter location address.
- When configuring PLC data tables, do not assign overlapping data addresses if floating point values are used and do not set an IEEEFP address to 1.

Data formats supported for Ethernet controllers

FSC types: BI, BO

Format	Description
S8B	Short signed integer
S16B	Signed word

Format	Description
S32B	Signed long integer
IEEEFP	IEEE Floating point

FSC types: AI, AO

Format	Description
FSC020MA	0 - 20 mA
FSC420MA	4 - 20 mA
FSC05V	0 - 5 V
FSC15V	1 - 5 V
FSC010V	0 - 10 V
FSC210V	2 - 10 V

Scaling with data formats

Parameter values with addresses in the Input register and Holding register data tables can be scaled with a data format. Select the format that corresponds to the counts that have been set in the PLC register.



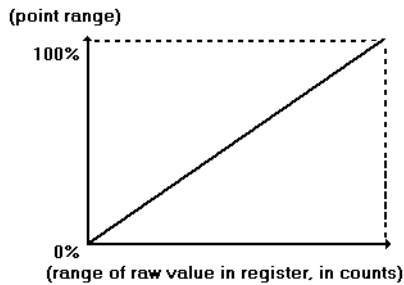
Note

If auxiliary parameters have a data format type that requires scaling (U4095, U999, and so on), they will take the same range as the PV.

The data format tells the server how to interpret the register value. Raw values in the PLC register tables can be scaled by the 0% and 100% point range values in order to convert them into engineering units (EU).

To select a format for scaling, you select the format that corresponds to the counts that have been set in the register where the point parameter value is sourced.

Figure 7 Scaling raw data



Location syntax for SOE only points

POS: *Index* SER *SerNumber*

Part	Description
<i>Index</i>	The server’s internal position allocated to this point. It must be unique. Range = 1 - 8192
<i>SerNumber</i>	The SER number. See “Assigning SER numbers for sequence of events” on page 19.

Optimizing scanning performance

The maximum amount of data that can be acquired from an FSC controller is influenced by the rate of sending scan packets to the controller. An understanding of FSC scan packets will help you configure points so that optimal data acquisition performance is achieved, by maximizing the amount of data acquired with each scan packet.

The scan packets that have been built can be listed by using the list scan utility, **lisscn**. Listing scan packets helps verify the scanning strategy. See the *Configuration Guide* for usage of **lisscn**.

Server and Station tasks for FSC

4

This chapter describes tasks for the FSC controller that you perform either on the server or from any Station.

For:	Go to:
Accessing diagnostics for an Ethernet controller	page 40
Testing communications with the server	page 41
Troubleshooting communication errors	page 42

Accessing diagnostics for an Ethernet controller

System information and extended diagnostic data are accessible from Station for FSC Ethernet controllers.

System information

To access FSC System Information, double-click the PV of an FSC Ethernet controller point. Alternatively, click **Details** in the Controller Status, which shows the **Ext Diagnostics** tab, then click the **System Info** tab for the required link.

The system information and extended diagnostic information is updated automatically every 50 seconds. You can manually refresh the information by clicking on the **Refresh** button.

If the diagnostic data cannot be retrieved from the FSC Ethernet controller, the most recent diagnostic data from the controller will continue to be displayed until the new data is obtained. If successive attempts to obtain diagnostic data from the controller fail, then an alarm message will be generated in the Alarm Summary.

Extended diagnostics

To access FSC Extended Diagnostics, click the **Details** button in the Controller Status. This shows up to 18 extended diagnostics messages for each FSC Ethernet controller link. Each message represents a hardware or software fault associated with that particular controller.

If an extended diagnostic message is present for a controller, an alarm with a value of `Message(s) Available` will be present in the Alarm Summary. To see the Extended Diagnostic information for an alarm from the Alarm Summary, double-click the alarm to call up the Controller Status, then click the **Details** button.

Testing communications with the server

Testing communications

Two diagnostic utilities, **fsctst** (for serial controllers) and **fsctst** (for Ethernet controllers), are included as part of the server software. These utilities test communications between the server and an FSC controller. Channels and controllers must be downloaded from Quick Builder before testing.

Before using the utility, make sure that:

- You have set up your FSC controllers according to their user manuals.
- All cables are connected.
- You have configured your channels in Quick Builder without error and downloaded all configuration information to the server without error.
- The server need not be running while using the utility as long as the database service is running. If making a connection through a terminal server, the server daemon service should also be running.
- The server is not communicating with your controllers. The test utilities might interfere with communications.

To stop the server, enter the command at the command prompt:

hscserver /load

Answer '**y**' to every prompt. This unloads the server, but leaves the database in memory.

Using fsctst

To use **fsctst**, start a command prompt window and type: **fsctst**.

When prompted for the channel number, enter **chn01** for channel 1 and so on.

For help using the utility, type **?**.

To check if your controllers are present, use the **find a,b** command. This command locates all FSC controllers on the channel with IDs between a and b.

For example:

```
C:\>fsctst
```

```
Enter LRN or device name of channel
```

```
chn01
```

```
Enter command:
```

```
find 1,4
```

```
FIND device with id 1 to 4, at 28-May-98 14:06:52
```

```
Device 1 ?
Device 2 ?
Device 3 ? ...responding
Device 4 ?
Enter command:
q
```

To the left of the channel name is the channel number. The name of the channel will be the letters “chn” followed by the two-digit channel number. For example, your Honeywell FSC channel “COM3” might be channel number 1. Its device name will be “chn01”.

Using fscetst

To use **fscetst**, start a command prompt window and type: **fscetst**.

The test utility is menu-driven, and self-explanatory. If there are any communications errors, appropriate error messages will describe what is wrong.

Troubleshooting FSC communication errors for a serial controller

If any errors are encountered, review the previous sections.

Error code 0106 (Device Timeout) means that no response was received from the controller.

Error code 8102 (MODBUS error 2 - illegal data address) means you either specified an illegal address or an illegal number of addresses.

After you have verified that the server is communicating with the controller, you can build points to reference controller addresses.



Note

Remember to enable the channel from the Channel Status display.

Troubleshooting FSC communication errors for an Ethernet controller

Ensure scan packets are optimized. For more information, see “Optimizing scanning performance” on page 37.

If there are several “logical” controllers on a channel, make sure that there is no more than one which has SOE enabled.

Index

A

- addressing
 - location syntax 32, 36
 - using Quick Builder 32
- architectures
 - Ethernet 13
 - serial 11

B

- Base SER 19

C

- channel, defining in Quick Builder 24
- communication address
 - Ethernet 17
 - serial 16
- communication settings 16
- communications, testing 41
- configuration utility 16, 17
- configuring FSC, steps for 5
- controller
 - addressing data tables 32
 - defining in Quick Builder 29

D

- data formats, scaling 35
- diagnostic utilities, using 41
- documentation for FSC 6

E

- Extended Diagnostics 40

F

- FSC System number

- Ethernet 17
- serial 16

- fsctst test utility, using 41

L

- LANVendor port, defining for FSC 28

P

- port definition
 - LANVendor 28
 - serial 26
 - terminal server 27
- Port tab 25

R

- redundancy
 - Ethernet 15
 - serial 13

S

- scanning, optimizing 37
- Sequence of Events 19
- SER number
 - described 19
 - reserved 19
- serial port, defining for FSC 26
- server tasks for FSC 39
- SOE
 - described 19
 - lines 20
- System Information 40

T

- terminal server port, defining for FSC 27
- time synchronization 31

INDEX

troubleshooting communication errors 42

Honeywell LCS620 Interface Reference

Issue

9.1

Date

November 2005

Notice

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 Limited Australia.

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.

Copyright 2005 – Honeywell Limited Australia

Honeywell trademarks

PlantScape[®], SafeBrowse[®], **TotalPlant**[®] and TDC 3000[®] are U.S. registered trademarks of Honeywell International Inc.

Experion[™] and Honeywell Enterprise Buildings Integrator[™] are trademarks 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.

Support and other contacts

For technical assistance, call your nearest Honeywell office.

Contents

1	Getting started	5
	Support and documentation for LCS620	6
	LCS620 models supported	6
	Other documentation	6
	LCS620-specific terms	7
2	LCS620 controller setup	9
	Connecting the LCS620 to the server	10
	Serial connection	10
	LAN connection	11
	Configuring the communications settings	13
	Configuring a serial connection	13
	Configuring a LAN connection	15
	Configuring a TDC Hiway connection	16
3	LCS620 controller configuration and addressing	17
	Defining an LCS620 channel	18
	LCS620 channel Main properties	18
	LCS620 channel Port properties	20
	LCS620 channel Redundant Port properties	22
	Defining an LCS620 controller	23
	LCS620 controller Main properties	23
	Defining an LCS620 address for a point parameter	25
	Entering an address	25
	Location syntax	25
	Optimizing scanning performance	28
	LCS620 packets	28
4	Server and Station tasks for LCS620	29
	Testing server to LCS620 communications	30
	Error messages	30

Getting started

1

This reference describes how to set up, configure, and test LCS620 controllers. There is detailed information for defining the controller using Quick Builder.

Steps for connecting and configuring an LCS620 Controller.

Complete each step before commencing the next step.

Step:	Go to:
Connect the LCS620 to the server	page 10
Set the LCS620 communication parameters	page 13
Define channels using Quick Builder	page 18
Define controllers using Quick Builder	page 23
Download channel and controller definitions to the server	
Test communications	page 30
Define controller points using Quick Builder	page 25

Support and documentation for LCS620

LCS620 models supported

The server supports the following LCS620 models:

- 620-6
- 620-11
- 620-12
- 620-14
- 620-16
- 620-20 to 620-35
- 620-36

Other documentation

The following LCS620 documents are relevant for configuring the LCS620:

- *LCS 620 documentation set*

The following documents contain design and configuration information:

- *Installation Guide*
- *Configuration Guide*

LCS620-specific terms

CIM

Communications Interface Module. Used to connect a controller to the server via a serial link.

EIM

Ethernet Interface Module. Required to connect a controller to the server via an Ethernet link (LAN).

LCS620 controller setup

2

This chapter describes how to set up an LCS620 controller.

The tasks for setting up an LCS620 controller are:

For:	Go to:
Connecting to the server	page 10
Communication settings	page 13

Connecting the LCS620 to the server

You can connect the LCS620 to the server using either serial, or LAN communications. The LCS620 supports redundant communication links.

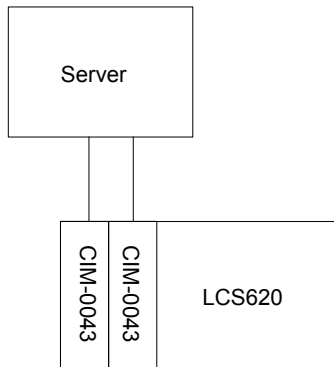
The server treats each serial or LAN link as a separate channel and each controller as a controller.

Serial connection

If you connect the LCS620 controller to the server with a serial connection, you can install a Communications Interface Module (CIM-0043) for each link. (The following figure shows a redundant link with two CIMs.)

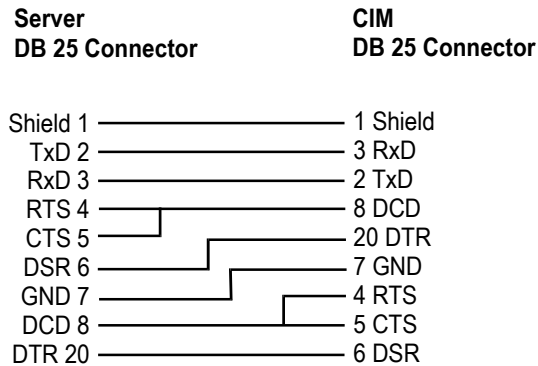
Alternatively, if you have model 12, 16 or 36 CPMs you can use their communications port. The LCS Loader software is required to configure this port. For more information, see the LCS Documentation.

You can use either RS-232 (point-to-point) or RS-422 (multi-dropped) protocols.

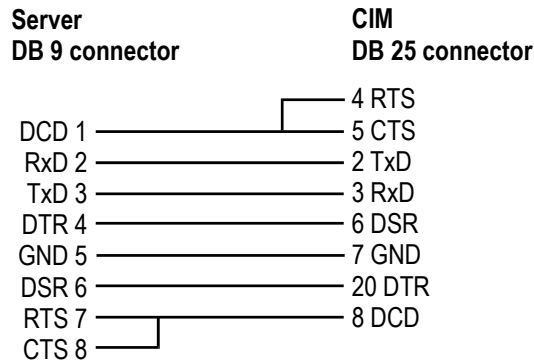


RS-232 connections

Use the following pinouts if the server has a 25-pin COM port.



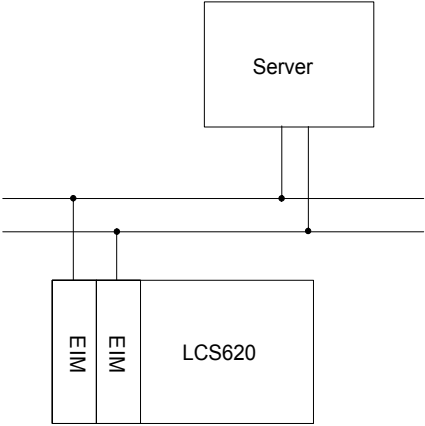
Use the following pinouts if the server has a 9-pin COM port.



These pinouts will also work with a Stallion Easyserver 2. Other terminal servers may require different handshaking options and this should be discussed with your local Honeywell representative.

LAN connection

If you connect the LCS620 controller to the server with an Ethernet connection, you must install an Ethernet Interface Module (EIM). (The following figure shows a redundant link with two EIMs.)



Configuring the communications settings

Before connecting the LCS620 to the server, configure the communications settings.



Note

Record the settings because you need them when using Quick Builder—see “LCS620 channel Port properties” on page 20.

Configuring a serial connection

The DIP switch settings in the CIM-0043 determine the serial link communications characteristics.

Switch bank (SW1)

SW1 controls the basic protocol settings: RS-232 or RS-422 (equivalent to RS-485).

Switch SW1	State	Switch function
1	Closed/On ⁱ	Resistor added for point-to-point connections with RS-422 interface
	Open/Off	Resistor removed for multidrop connections with RS-422 interface
2	Closed/On	Configures RS-422 transmitter for multipoint operation
	Open/Off ⁱ	Configures RS-422 transmitter for point to point operation, and CIM for full duplex operation.
3	Closed/On ⁱ	Selects RS-232
	Open/Off	Selects RS-422
4		Not used

ⁱ Recommended setting

Switch bank 2 (SW2)

SW2 controls the CIM's node address, which must be unique on the channel. If the controller has two CIMs (for a redundant connection), set both CIMs to the same address, but they must have different internal CIM addresses.

The internal CIM address is set by switches 7 and 8 on SW3. (For more information, see “Switch bank 3 (SW3)” on page 14.)

If there is only one controller on the channel, set the node address to 1.

Protocol restrictions place the following limitations on the node address:

- Zero is a broadcast node address. If you set the address to 0, the CIM will stall during its start-up self-test and cause the fault indicator to flash.
- Hardware limitations may hold the network population to less than 255 controllers.

The following table shows the address weighting of each SW2 switch, where the switch is used as part of the address if it is **Off** (open).

Switch SW2	Address weighting
1	1
2	2
3	4
4	8
5	16
6	32
7	64
8	128

The following table shows the settings for an address of 12.

Switch							
1	2	3	4	5	6	7	8
Closed	Closed	Open	Open	Closed	Closed	Closed	Closed

Switch bank 3 (SW3)

SW3, switches 1, 2, 3, and 4 control parity and stop bits.

Switch SW3	State	Switch function
1	Closed/On	CIM rejects Modify Coil Status, Modify I/O Register or Register commands
	Open/Off ⁱ	CIM accepts Modify Coil Status, Modify I/O Register or Register commands
2	Closed/On ⁱ	Parity Bit enabled
	Open/Off	Parity bit disabled
3	Closed/On	Odd parity
	Open/Off ⁱ	Even parity

Switch SW3	State	Switch function
4	Closed/On ⁱ	Selects one stop bit
	Open/Off	Selects two stop bits

i Recommended setting.

SW3, switches 5 and 6 control the baud rate.

Switch and state SW3		
5	6	Baud rate
Closed/On ⁱ	Closed/On ⁱ	19200
Open/Off	Closed/On	9600
Closed/On	Open/Off	4800
Open/Off	Open/Off	1200

i Recommended setting.

SW3, switches 7 and 8 control the CIMs address within the LCS620.

Switch and state SW3		Switch function
7	8	CIM address
Open/Off	Open/Off	3 ⁱ
Open/Off	Closed/On	2 ⁱ
Closed/On	Open/Off	1
Closed/On ⁱⁱ	Closed/On ⁱⁱ	0

i Only applies to 620-25 and 620-35.

ii Recommended setting.

Configuring a LAN connection

You need to download the configuration code to the EIM from the Series 9000 Configuration Tools. This loads the firmware into the EIM so that it can communicate.

You use the DIP switches in the EIM to set the IP address, which must be unique for all devices on the LAN. IP addresses are in Internet Class A, B or C format. The class is determined by your network requirements.

**Note**

The EIM does not support IP routing.

You must define the EIM's IP address in the TCP/IP host file on the server, C:\winnt\system32\drivers\etc\hosts, by adding the appropriate line. For example:

```
lcs620 200.0.0.50
```

In the case of a redundant communications link, where the controller has two EIMs, the alias of the first EIM should have an “a” suffix, and the alias of the second EIM should use the same alias, but with a “b” suffix, as shown in the following example:

```
lcs620a 200.0.0.50
```

```
lcs620b 200.1.1.51
```

If you define two logical controllers that address the same physical controller, you need to define the alias for each logical controller, for example:

```
logicalcont1 logicalcont2 200.0.0.50
```

Configuring a TDC Hiway connection

If you connect the LCS620 to a TDC Hiway via a Hiway Interface Module (HIM), you need to configure the interface as a TDC 3000 DHP.

For details about configuring the EIM, see the *Ethernet Interface Module User Manual*.

LCS620 controller configuration and addressing

3

This chapter describes how to configure an LCS620 controller using Quick Builder.

For information about:	Go to:
Defining an LCS620 channel	page 18
Defining an LCS620 controller	page 23
Defining an address for a point parameter value	page 25
Optimizing scanning performance	page 28

Defining an LCS620 channel

You must define each physical communication link to the LCS620 as an LCS620 channel.

To define a channel:

- 1 Click  to display the Add Items dialog.
- 2 Select `channel` as the item and `LCS620` as the type.
- 3 Set the property values on the Main tab—see “LCS620 channel Main properties” on page 18.
- 4 Click the Port tab and for **Port Type** select either:
 - `Serial`
 - `TerminalServer`
 - `LANVendor`See “LCS620 channel Port properties” on page 20.
- 5 If you need a redundant port:
 - a. Click the Redundant Port tab.
 - b. Select the same **Port Type** as for the primary port.
 - c. Set the other properties as appropriate—the redundant port settings must match primary port settings, except for **Port Name**.

LCS620 channel Main properties

The Main tab defines the basic properties for a channel.

Property	Description
Name	The unique name of the channel. A maximum of 10 alphanumeric characters (no spaces or double quotes).
Description	(Optional) A description of the channel. A maximum of 30 alphanumeric characters, including spaces.

Property	Description
Marginal Alarm Limit	The communications alarm marginal limit at which the channel is declared to be marginal. When this limit is reached, a high priority alarm is generated. A channel barometer monitors the total number of requests and the number of times the controller did not respond or response was incorrect. The barometer increments by 2 or more, depending on the error and decrements for each good call. To calculate an acceptable limit, multiply the square root of the number of controllers on the channel by the Marginal Alarm Limit defined for those controllers. (Normally, you specify the same value for all controllers on a channel). For example, if there are 9 controllers on the channel and their Marginal Alarm Limit is set to 25, the value would be $[\sqrt{9} \times 25] = 75$.
Fail Alarm Limit	The communications alarm limit at which the channel is declared to have failed. When this barometer limit is reached, an urgent alarm is generated. Set this to double the value specified in Marginal Alarm Limit.
Write Delay	(Only applicable to Serial port.) The time, in milliseconds, the server waits before writing to the controller.
Connect Timeout	The time (in seconds) the server waits to connect to the controller before abandoning the connection. The default value is 10 seconds. Use the default value unless the communications line has a high error rate or unless you are using modems.
Read Timeout	The time (in seconds) that the server waits for a reply from the controller. The default value is 2 seconds. Use the default value unless the communications line has a high error rate or unless you are using modems.
Item Type	Shows the type of item specified when this item was created.
Last Modified	Shows the date of the most recent modification to this channel's property details.
Last Downloaded	This shows the date that the item was last downloaded to the server.
Item Number	The unique item number currently assigned to this item. You can change the item number if you need to match your current server database configuration. The number must be between 1 and the maximum number of channels allowed for your system.

LCS620 channel Port properties

The Port tab defines the communication-related properties for a channel. The properties vary according to the selected **Port Type**:

- **serial**. Select this if you want to connect the controller to the server via a serial link. See “Serial port properties” on page 20.
- **terminalServer**. Select this if you want to connect the controller to the server via a LAN using a serial interface. See “Terminal server port properties” on page 21.
- **LANVendor**. Select this if you want to connect the controller to the server via a LAN using an Ethernet interface. See “LANVendor port properties” on page 22.



Note

Set the port properties to the same values as those specified when configuring the controller.

Serial port properties

Property	Description
Serial Port Name	The device name of the serial port.
Baud Rate	As per “Configuring a serial connection” on page 13
Number of Data Bits	As per “Configuring a serial connection” on page 13
Stop Bits	As per “Configuring a serial connection” on page 13
Parity	As per “Configuring a serial connection” on page 13
Checksum	As per “Configuring a serial connection” on page 13
XON/XOFF	As per “Configuring a serial connection” on page 13

Property	Description
RS-232	These properties are only applicable if controller is connected to the server with an RS-232 link. They are normally deselected for the LCS620. Enable RTS/CTS flow control. Select this if you want to use RTS/CTS for flow control to stop a receiver from being overrun with messages from a sender. Detect DCD. Select this if the Data Carrier Detect communication status line of the COM port requires monitoring (usually when using modem or microwave linking). If selected, the communications fails if the desired COM status line is not high, for example, on a dial-up link connection for a modem. Detect DSR. Select this if the Data Set Ready communication status line of the COM port requires monitoring (usually when using modem or microwave linking). If selected, the communications fails if the desired COM status is not achieved.
RS-485	These properties are only applicable if the controller is connected to the server with an RS-422 or RS-485 link. Select Enable Stallion RS-485 Half Duplex and Echo. (Echo indicates that the server expects messages it sends to the port on the transmit line to be echoed back on the receive line.)

Terminal server port properties

Property	Description
Terminal Server TCP Host Name	The name and port number of terminal server to which the channel is connected.
Terminal Server TCP Port No	You can specify either a TCP host name or an IP address, but it must match the TCP host name used when you installed and internally configured the terminal server.
Idle Timeout	The time, in seconds, the channel waits for a successful connection to the server before closing the connection. A value of 0 indicates that the connection is never closed.

Property	Description
Checksum	The type of checksum error detection used for the port. Select the value that matches the setting on the communication device: • CRC16_0 or CRC16_1 (if Cyclic Redundancy Check (CRC) is set) • ONESCOMP or TWOSCOMP (if Longitudinal Redundancy Check (LRC) is set) • XOR (If exclusive or is set)

LANVendor port properties

Property	Description
Port Name	Leave blank.

LCS620 channel Redundant Port properties

The Redundant Port tab defines the communication-related properties for a redundant channel. The redundant port parameter values are identical to the primary port values, except for **Port Name**.

Defining an LCS620 controller

You must define each LCS620 as a controller.

To define a controller:

- 1 Click  to display the Add Items dialog.
- 2 Select `Controller` as the item and `LCS620` as the type.
- 3 Set the property values on the Main tab. See “LCS620 controller Main properties” on page 23.

LCS620 controller Main properties

The Main tab defines the basic properties for a controller.

Property	Description
Name	The unique name of the controller. A maximum of 10 alphanumeric characters (no spaces or double quotes). If the controller is built on a LAN channel, you need to define this name as an alias in the TCP/IP host file. If the controller has a redundant LAN connection, you need a separate alias for each EIM. See “Configuring a LAN connection” on page 15.
Description	(Optional) A description of the controller. A maximum of 30 alphanumeric characters, including spaces.
Channel Name	The channel on which the controller communicates with the server.
Marginal Alarm Limit	The communications alarm marginal limit at which the controller is declared to be marginal. When this limit is reached, a high priority alarm is generated. This limit is the controller barometer, which monitors the total number of requests and the number of times the controller did not respond or response was incorrect. The barometer is incremented by 2 or more, depending on the error and decremented for each good call. The default value is 25.
Fail Alarm Limit	The communications alarm fail limit at which the controller is declared to have failed. When this barometer limit is reached, an urgent alarm is generated. Set this to double the value specified in Marginal Alarm Limit .

Property	Description
Node Address	Only applicable if the controller uses a serial connection. The node address, as set up in the CIM. See “Configuring a serial connection” on page 13.
Item Type	Shows the type of item specified when this item was created.
Last Modified	Shows the date of the most recent modification to this controller’s property details.
Last Downloaded	This shows the date that the item was last downloaded to the server.
Item Number	The unique item number currently assigned to this item. You can change the item number if you need to match your current server database configuration. The number must be between 1 and the maximum number of channels allowed for your system.

Defining an LCS620 address for a point parameter

Entering an address

For **PV Source Address**, **Source Address**, and **Destination Address** the format for an LCS620 controller address is:

ControllerName Location

Part	Description
<i>ControllerName</i>	The name of the LCS620 controller.
<i>Location</i>	The location in the controller where the value is stored. See “Location syntax” on page 25.

If you would like help when defining an address, click  next to **Address** to display Address Builder. For details, see the help.

Location syntax

The format for the location is:

Address [*DataFormat*|*BitNumber*]

Part	Description
<i>Address</i>	The address of the point parameter. See “Point parameter source/destination address ranges” on page 26. Note that LCS620 controllers have address restrictions that require relay ladder logic to facilitate access to parameters. See “Address restrictions” on page 26.
<i>DataFormat</i>	The data format acronym or starting bit number, depending on how you want the value to be read: • “Data format for scaling” on page 27 • “Data format for reading floating point values” on page 27 • “Data format for reading raw values without scaling” on page 27

Part	Description
<i>BitNumber</i>	The starting bit number for the point parameter that references a register in the PLC. The valid range is 0 (default) to 15 where 0 is the right-most bit in the register. The number of bits read is implied by the parameter type: analog parameters read up to 16 bits, status parameters read 1, 2 or 3 consecutive bits.

Example

Analog points

PV source: 5000

SP source: 5002 U9999

Status points

PV source: 2000

OP source: 6000 7

Point parameter source/destination address ranges

The address range varies according to the table type.



Note

Some models of LCS620 have smaller address ranges than those listed. Consult the LCS620 documentation for the precise address range of your particular model.

Table type	Address range	Point parameter		
		status	Analog	Accum
Real DI/DO	0 to 2047	PV, OP, MD	MD	
Internal DI/DO	2048 to 4095	PV, OP, MD	MD	
Register	4096 to 8191	PV, OP, MD	PV, OP, SP, MD, A1 to A4	PV

Address restrictions

Real analog Input/Output (0-2047) is not directly addressable.

Use relay ladder logic to move the analog value to and from a register.

Analog input cards are 12 bit; use data format U4095.

System status information, including redundancy status, is not directly addressable.

Use relay ladder logic to move this to internal I/O or registers for access.

LCS620 memory maps vary according to the model. See the LCS620 documentation for details.

Data format for scaling

Data formats for scaling are valid for analog points only. The default format is **U4095**.

Data Format	Counts
U4095	0 to 4095
U9999	0 to 9999
U999	0 to 999

Data format for reading floating point values

Select format **IEEEFP** to read two consecutive registers as a single precision floating point number.

Notes:

- IEEEFP numbers use two data addresses, **address** and **address-1**. Do not specify **address-1** (the lower address) as the point parameter location address.
- When configuring controller data tables, do not assign overlapping data addresses if floating point values are used and do not set an IEEEFP address to any value less than 4097.

Data format for reading raw values without scaling

Enter **0** to read all 16 bits in the register without scaling. To read less than 16 bits without scaling, enter the starting bit number (1 to 15).

Optimizing scanning performance

Minimize the number of scan packets—and the burden on the server—by using a small number of available scan periods for all your point definitions.

Example scan periods:

- 5 seconds for fast changing, important parameters
- 60 seconds for 1 minute PV history, parameters that change every few minutes
- 300 seconds for slow changing parameters

You verify your scanning strategy by using the List Scan utility, **lissen**, to list the scan packets you have built.

LCS620 packets

An LCS620 scan packet can consist of up to 122 16-bit words.

It can be located anywhere along the full address range of the controller.

To be in a scan packet, within the block, points must reside in the one controller and have the same scan period. In addition they must be addressed within the 122-word range that defines the block.

The scan packets can consist of up to 1952 digital addresses (both real and internal) and 122 registers. Registers can contain status, accumulator and analog points.

Two types of scan packet are built for an LCS 620 interface:

- **Hardware diagnostic scan.** One scan per controller every 60 seconds is initiated automatically by the server to verify communications integrity with the controller. One hardware diagnostic scan is automatically created per scan packet.
- **Periodic data acquisition scan.** On demand, the server acquires a value from a controller and processes the value as a point parameter.

For each source address scanned, a scan period can be specified. The period specified depends on the nature of the parameter. A fast changing or critical parameter uses a fast rate; an infrequent or non-critical (for example, SP) parameter should use a slower rate.

Server and Station tasks for LCS620

4

This chapter describes tasks for the LCS620 controller that you perform either on the server or from any Station.

For information about:	Go to:
Testing communications with the server	page 30

Testing server to LCS620 communications

You use the LCS620 diagnostic utility, **ipctst**, to test communications.

To run **ipctst** on the server:

- 1 At the server command line, type **ipctst** and press ENTER.
You are prompted to “Enter LRN or device name of channel”.
- 2 Type **chnxx** where xx is the channel number you want to test.
Note that the active LED on the CIM, EIM or ELPM flashes during communications.
- 3 To quit, type **q** and press ENTER.

To display help for **ipctst**, type **?** and press ENTER.

Error messages

0106 (device timeout)

No response was received from the controller.

If you have a serial connection (with a CIM), verify that the communication parameters defined in the project correspond with those set on the CIM and that the cable is correct.

If you have a LAN connection (with an EIM), verify that the Ethernet address in the TCP host file corresponds with those set on the EIM. Try using the TCP Ping command to verify the TCP connection to the EIM.

0110 (timeout on connect)

A physical connection to the LCS was not achieved within the configured period.

A possible cause of this error with the CIM (serial interface) is that it may be in Fault mode. The front of the CIM has a red LED to indicate Fault status. If the Fault condition recurs on each attempt to enable the channel, the problem is most likely due to incorrect pinouts. See to “RS-232 connections” on page 11 for the correct pinouts.

8425 (illegal PLC Address)

Ensure that the point parameter address lies within the correct range.

8426 (illegal PLC format)

Verify that the data format is a valid format for the address.

Index

C

- channel, defining 18
- CIM, described 10
- communication settings
 - LCS 620 13
- communications, testing 30
- configuring LCS 620, steps for 5
- connection
 - LAN 11
 - RS-232 11
 - RS-485 10
 - serial 10
 - server 10
- controller definition for a LCS 620 controller 23

D

- defining a LCS 620 address for a point
 - parameter 25
- documentation for LCS 620 6

E

- EIM
 - configuring 15, 16
 - described 11
- ELPM, described 11
- error messages 30
- Ethernet 11

I

- ipctst 30

L

- LAN connection
 - configuring 15
 - described 11

LCS 620

- communications settings 13
- connecting to a server 10
- models supported 6
- other documentation 6

M

- Main properties for a LCS 620 channel 18
- Main properties for a LCS 620 Controller 23
- models of LCS 620 6

P

- point parameter, defining an address 25
- Port properties for a LCS 620 20

R

- Redundant Port properties for a LCS 620
 - channel 22
- RS-232 connections 11

S

- scanning
 - optimizing performance 28
- serial connection
 - configuring 13
 - described 10
- serial port properties for a LCS 620 20
- server, connecting a LCS 620 to 10

T

- TDC Hiway 11
- TDC Hiway connection
 - configuring 16
- terminal server properties for a LCS 620 21

U

utility
 ipctst 30

Honeywell Micromax Interface Reference

Issue

4.1

Date

November 2005

Notice

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 Limited Australia.

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.

Copyright 2005 – Honeywell Limited Australia

Honeywell trademarks

PlantScape[®], SafeBrowse[®], **TotalPlant**[®] and TDC 3000[®] are U.S. registered trademarks of Honeywell International Inc.

Experion[™] and Honeywell Enterprise Buildings Integrator[™] are trademarks 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.

Support and other contacts

For technical assistance, call your nearest Honeywell office.

Contents

1	Getting started	5
	Support and documentation for the Micromax	6
	Devices supported	6
	Related documentation for Micromax controllers.	6
2	Micromax controller setup	7
	Supported architectures	8
	Connecting to the server	9
	Using an RS-232/422 converter	9
	Using the Stallion EasyConnection serial adapter.	10
	Communication settings	11
3	Micromax controller configuration and addressing	13
	Defining a Micromax channel.	14
	Micromax channel main properties.	14
	Micromax port properties	15
	Defining a Micromax controller	18
	Micromax controller main properties	18
	Defining a Micromax address for a point parameter value	20
	Entering an address	20
	Named access	20
	Hexadecimal access.	21
	I/O address numbering	22
	Value mapping between the server and Micromax	23
	Data formats	23
	Device addresses supported by the Micromax interface	24
	Interpreting integer data types.	28
	Micromax scanning.	32
	Troubleshooting Micromax configuration errors	33
	Point errors while downloading to the server	33
4	Server and Station tasks for Micromax	35
	Testing communications between the server and Micromax	36
	Troubleshooting Micromax scanning errors	37
	Configuring Micromax set point programs	38
	Viewing the set point program summary.	38
	Viewing or editing set point programs	39

CONTENTS

- Creating a new set point program 41
- Trending Micromax set point programs 41
- Configuring Micromax processes 43
- Configuring Micromax recipes 45
 - Viewing the Micromax recipe summary 45
 - Viewing and modifying a Micromax recipe configuration 46
 - Uploading and downloading recipes 47

Getting started

1

This reference describes how to set up, configure, and test Micromax controller communications with the server. There is detailed information for defining the controller using Quick Builder.

Complete each step before commencing the next.

Steps for connecting and configuring a Micromax Controller:

Steps:	Go to:
Connect the cables from the server to the Micromax controller	page 9
Set communication parameters	page 11
Define channels using Quick Builder	page 14
Define controllers using Quick Builder	page 18
Download channel and controller definitions to the server	
Test communications	page 36
Define controller points using Quick Builder	page 20

Support and documentation for the Micromax

Devices supported

The server supports many Honeywell Leeds and Northrup devices using this interface. Not all memory addresses exist on all the following devices:

- MICROMAX 1 LPU
- MICROMAX 2 LPU
- Speedomax 25000 Recorder
- RSX Videographic Recorder
- VPR100 Videographic Programmer Recorder
- CTX Controller

Related documentation for Micromax controllers

The following manual describes the address space of all the devices supported by this interface in great detail. It also provides information about the physical configuration of the units. This manual should be read when installing the interface.

- Honeywell - *Serial Communications User's Manual*; 51-52-25-54, Rev 0.0, 3/97. This manual has information relevant to the Micromax LPU, CTX, RSX, VPR 100, and Speedomax 25000.

The following manuals provide detailed information concerning the configuration of each type of unit:

- Honeywell - *Micromax 2 Field Station (LPU) Specifications and Installation Manual*; 70-MX-25-03 (replaces L&N Manual No. 277883)
- Honeywell - *MAXPRO Configuration Software (for LPU)*; 70-MX-25-13 (replaces L&N Manual 277877)
- Honeywell - *Progeny CTX Controller User's Manual*; Part No: 278600
- Honeywell - *Progeny RSX Video Recorder User's Manual*; Part No: 278606
- Honeywell - *VPR100 Video Programmer Recorder User's Manual*; 57-77-25-13, Rev 1.0, 6/97
- Leeds & Northrup - *Progeny RSX Video Recorder User's Manual*; Part No: 278606

Micromax controller setup

2

This chapter describes how to set up a Micromax for the system.

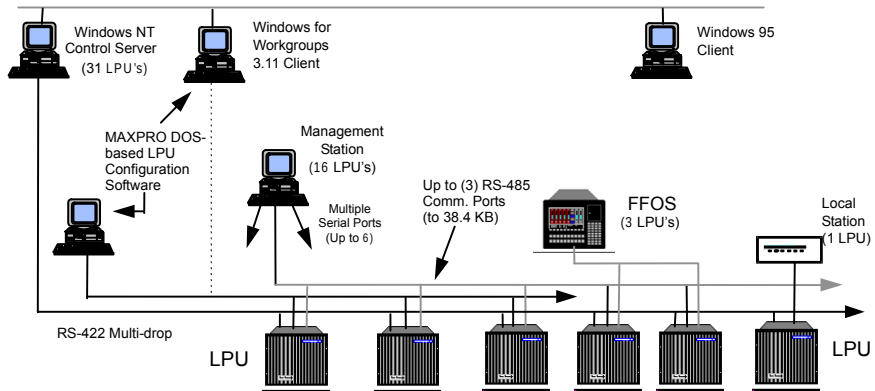
The tasks for setting up a Micromax controller are:

For:	Go to:
Supported Architectures	page 8
Connecting to the server	page 9
Communication settings	page 11

Supported architectures

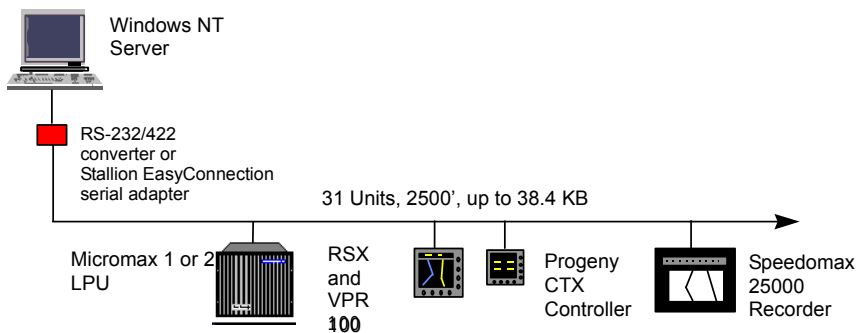
The following architecture is an example of a system design integrating the server with a network of Micromax LPUs.

Figure 1 Sample system design



The following is the architecture for a single channel consisting of a mixture of product types supported by the interface.

Figure 2 Single-channel mixed architecture

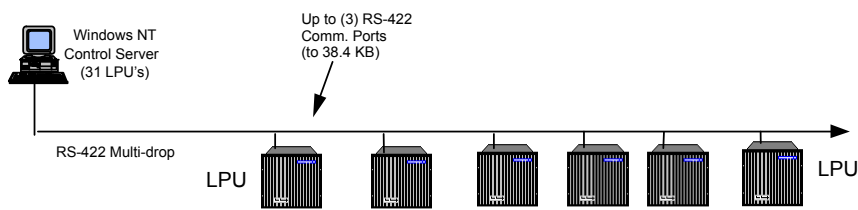


Connecting to the server

Connect the server to the Micromax over a serial line using RS-422. In order for the server to use RS-422, either an external RS-232/422 converter must be used on a serial port, or internal RS-422 must be supported on the server (via the Stallion EasyConnection Adapter).

The RS-422 server interface to the Micromax allows up to 31 units to be connected in a multi-dropped configuration.

Figure 3 Micromax RS-422 Multi-drop Connection



You can use any of the available serial ports of the Micromax LPU. The LPU has up to three in newer systems with the 20 MHZ CPU card, two in older systems typically using the 10 MHz or 6 Mhz CPUs. All other hardware types have only one serial port.

Use either COM1 or COM2 of the computer normally, dependent on availability. Another alternative is to use the Stallion EasyConnection serial adapter. All units must communicate at the same baud rate on the same multi-dropped channel.

“Micromax RS-422 Multi-drop Connection” on page 9 shows the hardware connection recommended for the RS-232/422 converter and the Stallion EasyConnection serial adapter.

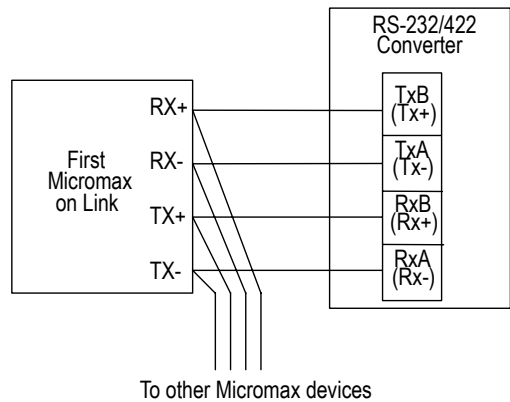
Using an RS-232/422 converter

Honeywell recommends that you use the Black Box LD422-HS RS-232/422 Interface Converter, model number ME837A. This converter has been qualified by Honeywell. Use of another converter may produce unexpected results.

Connect the Micromax, or other products supported by this interface, to the Black Box converter as shown in “Controller cable connection” on page 10.

Note that the last unit in the communications link must be terminated and all other slave units must be unterminated, or data transfer will not function properly. See the appropriate device’s User’s Manual for instructions on enabling/disabling the termination resistors.

Figure 4 Controller cable connection

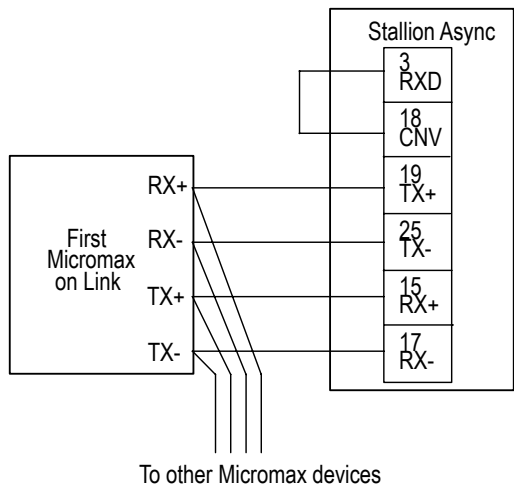


Using the Stallion EasyConnection serial adapter

If the Stallion EasyConnection serial adapter has not been installed in the server, see the installation instructions in the *Installation Guide*.

Connect the Micromax, or other products supported by this interface, to the Stallion EasyConnection async module as shown in “Stallion RS-422 controller cable connection” on page 10.

Figure 5 Stallion RS-422 controller cable connection



Communication settings

The correct communication parameters are required to communicate with the server. See the Micromax documentation for details on setting up the following parameters:

Unit address

The unit address of each device on the network must be set to a unique value between 1 to 254.

Parity

Parity is selectable on the Micromax units. Although use of parity is recommended, parity may be odd, even, or none.

Baud rates

Baud rates are selectable for each type of unit. All units must communicate at the same baud rate on the same multi-dropped channel. All can communicate up to 38.4 KB except the Speedomax 25000 and older 6 or 10 Mhz Micromax LPUs which are limited to 19.2 KB. The following table displays selectable baud rates for each of the supported devices:

Table 1 Baud Rates

BAUD Rate	SPEEDOMAX 25000	MICROMAX LPU	PROGENY
38400		Yes	Yes
19200	Yes	Yes	Yes
9600	Yes	Yes	Yes
4800			Yes
2400			Yes
1200	Yes	Yes	Yes

Terminating resistor

Note that the last unit in the communications link must be terminated and all other slave units must be unterminated, or data transfer will not function properly. See the appropriate device's User Manual for instructions on enabling or disabling the termination resistors.

Micromax controller configuration and addressing

3

This chapter describes how to configure a Micromax controller for the system using Quick Builder. For each configuration procedure, there is detailed information covering all supported Micromax devices.

This chapter also contains reference information about supported Micromax addresses, I/O mapping, interpreting integer data types, and value mapping.

The following Quick Builder configuration tasks are described:

For:	Go to:
Steps for defining a Micromax channel	page 14
Steps for defining a Micromax controller	page 18
Defining an address for a point parameter value	page 20
Address syntax for Micromax addresses supported by the server	page 20
Address syntax for Micromax addresses not supported by the server	page 22
How to optimize scanning performance	page 32
Troubleshooting downloading errors	page 33

Defining a Micromax channel

You need to define a channel for each server connection to a Micromax controller.

To define a channel using Quick Builder:

- 1 Click  to display the Add Items dialog box.
- 2 Select Channel as the item and Micromax as the channel type.
- 3 Complete the Micromax channel definition on the Main property page for the channel. For help with the channel definitions, see “Micromax channel main properties” on page 14.
- 4 When you have finished the Main channel definitions, click the Port tab to display the Port properties for a channel and define the port type. See “Micromax port properties” on page 15.

Micromax channel main properties

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

Property	Description
Name	Type a unique name for the channel. A maximum of 10 alphanumeric characters can be used. (Double quotation marks and spaces are not allowed.)
Description	(Optional) Type a description of the channel. A maximum of 30 characters can be used, including spaces.
Marginal Alarm Limit	Communications alarm marginal limit at which the channel is declared to be marginal. When this limit is reached, a high priority alarm is generated. A channel barometer monitors the total number of requests and the number of times the controller did not respond or response was incorrect. The barometer increments by 2 or more, depending on the error and decrements for each good call. To calculate an acceptable limit, multiply the square root of the number of controllers on the channel by the Marginal Alarm Limit defined for those controllers. (Normally, you specify the same value for all controllers on a channel). For example, if there are 9 controllers on the channel and their Marginal Alarm Limit is set to 10, the value would be $[\sqrt{9} \times 10 = 30]$.

Property	Description
Fail Alarm Limit	Communications alarm fail limit at which the channel is declared to have failed. When this barometer limit is reached, an urgent alarm is generated. Set this to double the value specified for the channel Marginal Alarm Limit .
Item Type	Shows the type of item specified when this item was created.
Last Modified	Shows the date of the most recent modification to this channel's property details.
Item Number	This field displays the unique item number currently assigned to this item by Quick Builder. You can change the item number displayed in this field if you need to match your current server database configuration. The item number must be between 1 and the maximum number of channels allowed for your system.

Micromax port properties

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

- **serial**. A serial communications interface, such as RS-232. See “Serial port properties” on page 15.
- **terminalserver**. A communications link that enables controllers with a serial interface to be connected to a LAN. See “TerminalServer port properties” on page 16.
- **lanvendor**. Not used with Micromax.

Serial port properties



Note

The Serial Port settings must match the settings on your communication devices.

Property	Description
Serial Port Name	The device name of the serial port.
Baud Rate	The number of data bits per second. The default is 9600.
Number of Data Bits	The number of data bits used for transmission. The default is 8.

Property	Description
Stop Bits	The number of stop bits used for transmission. The default is 1.
Parity	The parity verification used on the port. The default is NONE.
Checksum	Not applicable for Micromax.
XON/XOFF	Not applicable for Micromax.
RS-232	Only applicable for a an RS-232 link. (The RS-232 and RS-485 settings are mutually exclusive.) Enable RTS/CTS flow control. Select this if you want to use RTS/CTS for flow control to stop a receiver from being overrun with messages from a sender. Detect DCD. Select this if the Data Carrier Detect communication status line of the COM port requires monitoring (usually when using modem or microwave linking). When selected, the communications fails if the desired COM status line is not high—for example, on a dial-up link connection for a modem. Detect DSR. Select this if the Data Set Ready communication status line of the COM port requires monitoring (usually when using modem or microwave linking). When selected, the communications fails if the desired COM status is not achieved.
RS-485	Only applicable for a an RS-485 link. (The RS-485 and RS-232 settings are mutually exclusive.) Enable Stallion RS-485 Half Duplex. Select this if you are using a Stallion EasyConnection. Don't select this if you use a 4-wire Stallion RS-422 connection. Echo. Select this to indicate that the server should expect the messages it sends to the port on the transmit line to be echoed back on the receive line. For example, select this if you are using a Stallion EasyConnection adapter. For RS-485 connections using a Black Box Converter, leave the box unselected (no echo).

TerminalServer port properties

Property	Description
Terminal Server TCP Host Name	The name and port number of terminal server to which the channel is connected.

Property	Description
Terminal Server TCP Port No	You can specify either a TCP host name or an IP address, but it must match the TCP host name used when you installed and internally configured the terminal server.
Idle Timeout	The time, in seconds, the channel waits for a successful connection to the server before closing the connection. A value of 0 indicates that the connection is never closed.
Checksum	Not applicable for Micromax.

Defining a Micromax controller

You need to define a controller to describe each Micromax LPU on a channel.

To define a controller with Quick Builder:

- 1 Click  to display the Add Items dialog box.
- 2 Select Controller as the item and Micromax as the controller type.
- 3 On the Main tab for the Micromax controller, complete the controller definition.
See “Micromax controller main properties” on page 18 for help with the controller definitions.

Micromax controller main properties

Use the Main tab to name the controller and define its basic properties.

Property	Description
Name	Type a unique name for the controller. A maximum of 10 alphanumeric characters can be used. (Double quotation marks and spaces are not allowed.)
Description	(Optional) Type a meaningful description for this controller. A maximum of 30 characters, including spaces, can be entered.
Channel Name	The name of the channel on which the controller communicates. In the list of channel names, click the name. You need to have defined the channel in order for it's name to appear in the list.
Marginal Alarm Limit	The communications alarm marginal limit at which the controller is declared to be marginal. When this value is reached, a high priority alarm is generated. This limit applies to the controller barometer which monitors the total number of requests to the controller and the number of times the controller did not respond or response was incorrect. The barometer increments by 2 or more, depending on the error and decrements for each good call. Type a new value in this field if you do not want the default, 25.

Property	Description
Fail Alarm Limit	The communications alarm fail limit at which the controller is declared to have failed. When this barometer value is reached, an urgent alarm is generated. Set this to double the value specified for the controller Marginal Alarm Limit.
Controller Type	Specify a type for the controller, either: • default Use for all devices that do not use a 10 MHz CPU card, including most Micromax LPU systems, Progeny CTX, RSX, and Speedomax devices. • Micromax 2 10Mhz CPU Specify when you need to use the Micromax LPU with the 10 MHz CPU card. Adds a 60ms delay between writes to the controller. This might be required for some older models of Micromax as the 10 MHz card requires the protocol to insert a delay between I/O requests.
Controller ID	Type the controller unit address specified on the Micromax device.
Item Type	Shows the type of item specified when this item was created.
Last Modified	Shows the date of the most recent modification to this channel's property details.
Item Number	This field displays the unique item number currently assigned to this item by Quick Builder. You can change the item number displayed in this field if you need to match your current server database configuration. The number must be between 1 and the maximum number of controllers allowed for your system.

Defining a Micromax address for a point parameter value

Entering an address

For **PV Source Address**, **Source Address**, and **Destination Address** the format for a Micromax controller address is:

ControllerName Location

Part	Description
<i>ControllerName</i>	The name of the Micromax controller.
<i>Location</i>	The location in the controller where the value is recorded. Location syntax can be entered as either: • “Named access” on page 20 (for supported addresses) • “Hexadecimal access” on page 21 (for unsupported addresses)

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

Named access

The location syntax for a supported address is:

ParameterType### [BitNumber | Format]

Part	Description
<i>ParameterType</i>	ASCII representation (name) of the parameter type. See “Device addresses supported by the Micromax interface” on page 24.
<i>###</i>	The I/O address number in decimal. See “I/O address numbering” on page 22.
<i>BitNumber</i>	(Optional) For status point addresses, a number between 0 and 31. See “Interpreting integer data types” on page 28. Bits can be read from valid bit offsets within the addresses range from 0 to 31.
<i>Format</i>	(Optional) Specify a format if the address holds a floating point value. For more details see “Data formats” on page 23.

Example SYSTAT (system status as an integer)
 SYSTAT 4 (Run/Online mode of the LPU)
 LPV10 (Loop Process Variable for loop 10)
 DO001 (DO status value for DO #1).

Hexadecimal access

A large number of Micromax parameters do not have an ASCII name but can still be addressed by the Micromax interface. A second addressing method, hexadecimal, is used to specify these addresses.



Caution

The Micromax interface should work with most of these parameters, however, it has not been tested with all of them and its operation is not guaranteed.

The location syntax for a hexadecimal address is:

`0xTTAA [BB | Format]`

Part	Description
<i>TT</i>	Hexadecimal representation of the parameter type. See the <i>Serial Communications User's Manual For Leeds & Northrup Instrument Protocol</i> for a full description of available parameter types and addresses. Note: An address specified in hexadecimal format for which a supported ASCII address exists is converted to the ASCII format when a server utility, such as bckbld or lissen , is run.
<i>AA</i>	The I/O address number in decimal. See "I/O address numbering" on page 22.
<i>BB</i>	(Optional) For status point addresses, a number between 0 and 31. See "Interpreting integer data types" on page 28. Bits can be read from valid bit offsets within the addresses range from 0 to 31.
<i>Format</i>	(Optional) Specify a format if the address holds a floating point value. For more details see "Data formats" on page 23.

Example 0x0200 (System status as an integer.)
 0x0200 4(Run/Online mode of the LPU)
 0x030a (Loop Process Variable for loop 10)
 0x2301 (DO status value for DO #1)

I/O address numbering

Device	Address is	Go to:
Micromax LPU	I/O numbering is based on the position of the input or output in the rack relative to other inputs or outputs of the same type.	page 22
Speedomax and other devices	I/O numbering is based on the absolute position of the input or output in the rack.	page 22
Progeny	The I/O numbering is the same as that observed using the device faceplate.	

Relative addressing

Inputs and outputs are numbered by their position in the card racks relative to other inputs/outputs of the same type.

Example If the I/O cards installed in a Micromax, LPU are:

Cards	I/O ranges for each type
8 input AI card in position 1	AI1 to 8
8 input AO card in position 2	AO1 to 8
15 input DI card in position 3	DI1 to 15

Absolute addressing

I/O numbering depends on the card's position in the rack. Each card position uses up 15 I/O numbers, regardless of the number of I/O installed in that position.

Example The same I/O cards (as used in the example for Relative addressing) installed in a Speedomax are:

Cards	I/O ranges for each type
8 input AI card in position 1	AI1 to 8
8 input AO card in position 2	AO16 to 23
15 input DI card in position 3	DI31 to 45

Value mapping between the server and Micromax

When accessing certain parameter types in the Micromax, the point parameter value in the server is mapped to a different value in the Micromax. For example, to enable a digital value to action a “Force-ON” when writing to a DI parameter in the Micromax, a server value of 3 is converted to a value of 129 before writing to the device. The reverse mapping also applies.

Parameter Type	Server Value	Micromax or Progeny Value	Action
DI Digital Input (0x21)	0	0	Off (releases any Force)
	1	1	On (releases any Force)
	2	128	Force-Off
	3	129	Force-On
DO Digital Output (0x23)	0	0	Off (releases any Force)
	1	1	On (releases any force)
	2	128	Force-Off
	3	129	Force-On

Data formats

Any address of data type “int” that holds an integer value requires the U32B format (an unsigned 32-bit integer). The U32B format is the default and therefore it is not necessary to specify a data format for an integer address in Quick Builder.

Any address of data type “float” that holds a floating point value in the controller requires the IEEEFP format.

Data Format	Description	Scaled
U32B (Default)	Data is interpreted as an unsigned 32-bit integer. If the data read is less than 32-bits, remaining bits will be set to 0.	No
IEEEFP	IEEE Floating Point number	No

Device addresses supported by the Micromax interface

For some addresses, data conversion between the server and Micromax is done. For more information on data conversion, see “Interpreting integer data types” on page 28.

Parameter Description	Named Address	Param Type Hexadecimal?	Data Type	Data Conversion	Read/Write	UNIT Code ⁱ
AI: Range High	aihi1-225	0xBB??	float	No	R	S,M,P
AI: Range Low	ailo1-225	0xBA??	float	No	R	S,M,P
AI: Status	ais1-225	0x9C??	int	Yes	Read	S,M,P
AI: Adjustment Value	aivadj1-225	0x15??	float	No	R/W	S,M,P
AI: Output Value	aiv1-225	0x07??	float	No	Read	S,M,P
AL: Output Status	alos1-150	0x12??	int	No	Read	S,M,P
AL: Alarm setpoint	alsp1-150	0x11??	float	No	R/W	S,M,P
AO: Output value	aov1-16	0x3E??	float	No	read	S,M,P
LP: Approach Value High	avhi1-255	0x0F??	float	No	R/W	M,P
LP: Approach Value Low	avlo1-255	0x10??	float	No	R/W	M,P
Local Time	clock	0x1D00	float	Yes	R/W	S,M,P
CN: Constant value	cnv1-200	0x25??	float	No	R/W	S,M,P
CR: Control relays output	cr1-255	0x7B??	int	Yes	read	M
CR: Block 2	cr_b0-255	0x7C??	int	No	read	M
CR: Block 3	cr_c0-255	0x7D??	int	No	read	M
CR: Block 4	cr_d0-231	0x7E??	int	No	read	M
DI: Status	di1-225	0x21??	int	Yes	R/W	M,P

Parameter Description	Named Address	Param Type Hexadecimal?	Data Type	Data Conversion	Read/Write	UNIT Code ⁱ
Diagnostic error status	diagst1-44	0x14??	int	No	read	S,M
DO: Status	do1-225	0x23??	int	Yes	R/W	M,P
LP: Control Auto/Man selection	lams1-16	0x56??	int	No	write	M
LP: Control Auto/Man selection		0x56??	int	No	R/W	P
LP: Control Auto/Man status	lamst1-16	0x55??	int	No	read	M
LP: Control Auto/Man status		0x55??	int	No	R/W	P
LP: Bias primary	lbiasp1-16	0x62??	float	No	R/W	M,P
LP: Bias secondary	lbias1-16	0x68??	float	No	R/W	M
LP: Carbon; CO compensation	lccol-16	0x98??	float	No	R/W	M
LP: Carbon; furnace factor	lcf1-16	0x99??	float	No	R/W	M
LP: Control output force manual	lcotm1-16	0x57??	int	No	read	M,P
LP: Carbon; Sensor	lcs1-16	0x96??	float	No	read	M
LP: Carbon; Furnace temp	lct1-16	0x97??	float	No	read	M
LP: Deviation	ldev1-16	0x06??	float	No	read	M,P
LP: Gain primary	lgainp1-16	0x0B??	float	No	R/W	M,P
LP: Gain secondary	lgains1-16	0x09??	float	No	R/W	M,P
LP: Local setpoint	llsp1-16	0x04??	float	No	R/W	M,P
LP: Man out value	lmo1-16	0x3C??	float	No	R/W	M,P
LP: Man out primary	lmop1-16	0x69??	float	No	R/W	M,P
LP: Man out secondary	lmos1-16	0x6A??	float	No	R/W	M
LP: Manual reset	lmr1-16	0x33??	float	No	R/W	M

Parameter Description	Named Address	Param Type Hexadecimal?	Data Type	Data Conversion	Read/Write	UNIT Code ⁱ
LP: Bias (P: Manual reset)		0x33??	float	No	R/W	P
LP: Force off select	loof1-16	0x65??	int	No	R/W	M
LP: Output ON/OFF status	loos1-16	0x66??	int	No	read	M
LP: Output ON/OFF status		0x66??	int	No	R/W	P
LP: Output value	lov1-16	0x08??	float	No	read	M
LP: Output value		0x08??	float	No	R/W	P
LP: Process variable	lpv1-16	0x03??	float	No	read	M,P
LP: Rate secondary	lrats1-16	0x3A??	float	No	R/W	M,P
LP: Ratio	lratio1-16	0x6C??	float	No	R/W	M,P
LP: Rate primary	lratp1-16	0x0D??	float	No	R/W	M,P
LP: Remote setpoint	lrsp1-16	0x05??	float	No	R/W	M,P
LP: Reset secondary	lrsts1-16	0x0A??	float	No	R/W	M,P
LP: Reset primary	lrstp1-16	0x0C??	float	No	R/W	M,P
LP: Preset output	lsop1-16	0x36??	float	No	R/W	M
LP: Working setpoint	lsp1-16	0x2D??	float	No	read	M,P
AO: Failsafe/ Preset value		0x36??	float	No	R/W	P
LP: Set point selection	lspsl1-16	0x53??	int	No	write	M
LP: Set point selection		0x53??	int	No	R/W	P
LP: Set point slew rate	lspsr1-16	0x30??	float	No	R/W	M
LP: Decreasing slew rate limit		0x30??	float	No	R/W	P
LP: Set point status	lspst1-16	0x52??	int	No	read	M
LP: Set point status		0x52??	int	No	R/W	P

Parameter Description	Named Address	Param Type Hexadecimal?	Data Type	Data Conversion	Read/ Write	UNIT Code ¹
LP: Tuning parameter select	ltunsl1-16	0x63??	int	No	R/W	M,P
LP: Wild variable	lwild1-16	0x0E??	float	No	R	M,P
Operation Status	opstat	0x0220	int	No	Write	S,M
Operation Status		0x0220	int	No	R/W	P
PLC: Counter preset value	plccrst1-80	0x82??	float	No	R/W	M
PLC: Timer current value	plctv1-80	0x7F??	float	No	read	M
PLC: Timer preset	plctset1-80	0x80??	float	No	R/W	M
PLC: Counter current value	plccv1-80	0x81??	float	No	read	M
PLC: Drum step #	plcdrum1-4	0x83??	float	No	R/W	M
PLC: Drum step time	plcdstm1-4	0x84??	float	No	R/W	M
PLC: Drum outputs (packed)	plcdop	0X7101	int2	No	read	M
PP: Output low limit	pplo1-255	0x4B??	float	No	read	S,M
PP: Output low limit		0x4B??	float	No	R/W	P
PP: Output high limit	pphi1-255	0x49??	float	No	read	S,M
PP: Output high limit		0x49??	float	No	R/W	P
PP/CV: Pseudo Point	ppv1-250	0x1E??	float	No	Read	S,M,P
System Status	systat	0x0200	int4	No	Read	S,M,P
SPP: Output value	sppov1-4	0x4D??	float	No	read	M,P
SPP: Elapsed time	sppet1-4	0x4E??	float	No	read	M,P
SPP: Time remaining	spprt1-4	0x4F??	float	No	read	M,P
SPP: Step number	spps1-4	0x50??	float	No	read	M,P
SPP: Start	sppstr1-4	0x73??	int	Yes	write	M,P
SPP: Hold	spphld1-4	0x75??	int	Yes	write	M,P

Parameter Description	Named Address	Param Type Hexadecimal?	Data Type	Data Conversion	Read/Write	UNIT Code ⁱ
SPP: Advance	sppadv1-4	0x76??	int	Yes	write	M,P
SPP: Reset	spprst1-4	0x74??	int	Yes	write	M,P
SPP: Jog	sppjog1-4	0x77??	int	Yes	write	M,P
SPP: Status	sppsts1-4	0x78??	int	No	read	M,P
SPP: Step time	sppst1-16	0x8A??	float	No	R/W	P
SPP: Step value	sppsv1-16	0x88??	float	No	R/W	P
TL: Totalizer value	tlv0-6	0x1C??	float	No	read	P

- i The unit codes are:
M = Micromax, such as MICROMAX 1 LPU
S = Speedomax, such as Speedomax 25000 Recorder
P = Procency, such as CTX Controller

Interpreting integer data types

systat

Bit	Meaning
0	offline
1	offline diag
2	offline calib
3	diag hold / maint hold
4	run / online mode
5	cold start
6	warm start
7	program mode
8	first time power up
9	spare
10	data storage present
11	RAM board present
12	ROM board present
13	comm board present
14	PID present
15	PLC present

Bit	Meaning
16	ESC pressed
17	+/- pressed
18	ENTER pressed
19	50 Hz operation
20	reserved
21	reserved
22	reserved
23	operator MMI active
24	program MMI active
25	spare
26	spare
27	spare
28	spare
29	comm B present
30	comm A present
31	printer present

opstat

Conversions for Micromax LPU only. See the Leeds & Northrup manual for other devices and tables.

Bit	Meaning
0	offline diag
1	offline calib
2	offline
3	program
4	cold start
5	online
6	spare
7	spare

diagst0

Bit	Meaning
0	powerup
1	calib area error
2	calc area error
3	user prog area err
4	unspecified error
5	Real-time clock error
6	
7	low battery

ais

Bit	Meaning
0	open input status (1=reported)
1	AI comp/emmiss (1=active)
2	underrange (1=under)
3	overrange (1=over)
4	point skip (1=skip)
5	open status (1=open)
6	open check (1=enable)
7	deg F/C (1=deg F)

sppsts

Bit	Meaning
0,1	0=stop, 1=run, 2=delayed start, 3=hold
2	1=start
7	time units; 0=minutes, 1=seconds
0,1	0=stop, 1=run, 2=delayed start, 3=hold

ltunsl

0=primary
1=secondary

lamst

0>manual

1=auto

lamsl

0>manual

1=auto

lcotm

0=force manual

lspsl

0=local (SP1)

1=remote(SP2)

lspst

0=local (SP1)

1=remote(SP2)

loos

0=off

1=on

loof

1=force off

Micromax scanning

Reducing the number of scan packets

If the number of scan packets becomes too great, scanning performance will be impaired.

Up to 100 points using the same scan period can be combined into a single scan packet. To minimize the number of scan packets, use a small number of available scan periods for all your point definitions.



Note

The scan packets that have been built can be listed by using the utility **lisscn** (list scan). See the *Configuration Guide* for usage of **lisscn**.

Troubleshooting Micromax configuration errors

Point errors while downloading to the server

If points are configured with details that are not valid, this can cause problems when the points are downloaded to the host. If problems do occur during the download, read the output file created and correct the errors.

You can view the Quick Builder Download Error output file either at download time or afterwards. See [Viewing the Download Error Log](#) in the Quick Builder help.

Server and Station tasks for Micromax

4

This chapter describes tasks for the Micromax controller that you perform either on the server or from any Station.

For:	Go to:
Testing communications with the server	page 36
Troubleshooting scanning errors	page 37
Configuring Set Point programs	page 38
Configuring Micromax Processes	page 43
Configuring Micromax Recipes	page 45

Testing communications between the server and Micromax

Use the Micromax test utility, **umaxtst**, to test communications between the server and the controller after you have downloaded your channel and controller definitions to the server database.

Before testing:

- Set up controllers.
- Connect all cables.
- Enter definitions for controller and channel in Quick Builder project file.
- Download project file definitions to server without errors.

To run the test utility, on the operating system command line type **umaxtst** and press ENTER.

Follow the directions as prompted.

Troubleshooting Micromax scanning errors

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

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

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

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

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

Configuring Micromax set point programs

If you use Micromax LPU controllers you can use Station displays to view or edit Set Point Programs or Recipes. You will still need to configure the Set Point Programs for the controller using the MAXPRO Configuration software. See the MAXPRO documentation for details.

To:	Go to:
View the Set Point Program Summary	page 38
Edit Set Point Programs	page 39
Create a new Set Point Program	page 41
Create Trend displays for Set Point Programs	page 41

Viewing the set point program summary

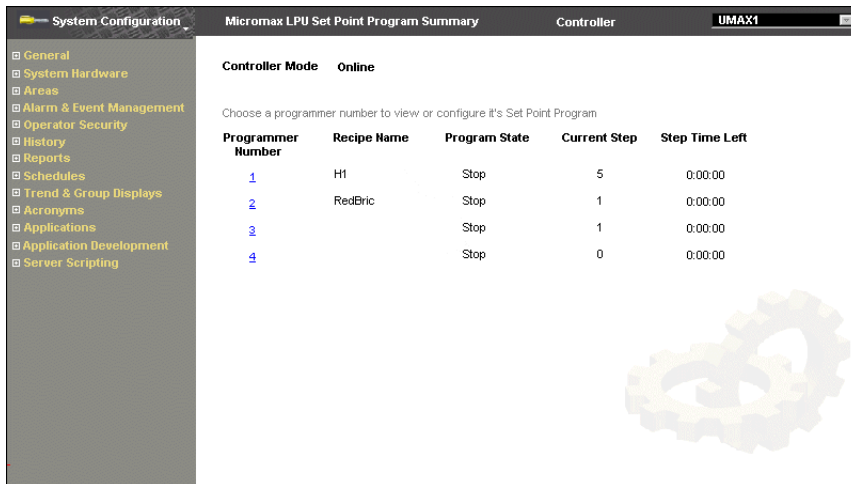
Each Micromax controller can contain up to four Set Point Programs.

To view a summary of a particular controller's Set Point Programs:

- 1 Select **Configure > Applications ⇒ Micromax ⇒ Set Point Programs**.

This calls up the Micromax Set Point Program Application Summary display.

Figure 6 Micromax Set point program summary display



- From the **Controller** list in the title bar, select the controller whose Set Point Programs you want to view or edit. The list displays all the controllers on Micromax channels that are Micromax LPUs.

All the Micromax Set Point Programs are read from the selected controller and the following information is displayed:

- Recipe name (if SPP has been downloaded from a recipe)
- Status of the Set Point Program
- Current step that the Set Point Program is on
- Time remaining for current step to complete
- Current controller node

Viewing or editing set point programs

To view details of a Set Point Program:

- On the Micromax LPU Set Point Program Summary display, click the **Programmer Number** of the programmer running the Set Point Program you want to display.

This calls up the Set Point Program Configuration display for that programmer.

Figure 7 Set Point Program Configuration Display

System Configuration | Micromax LPU Set Point Program | Controller: UMAX1 | Programmer: 1

Length: 5 Step | Step Time Left: 0:00:00 | Starting Time: 0:00:00 | Output Value: 5.0000

Recipe Name: H1 | Elapsed Time: 0:05:58 | Program State: Stop

Step	Type	Start Value (EU)/ Soak Value (EU)	Rate (EU/min)/ Soak Time (min)	Events
1	Soak	0.00	1.00	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
2	Ramp	0.00	1.00	
3	Soak	1.00	1.00	
4	Ramp	1.00	2.00	
5	Soak	5.00	1.00	
0		0.00	0.00	
0		0.00	0.00	
0		0.00	0.00	
0		0.00	0.00	
0		0.00	0.00	

Setting the Type to blank will clear remaining segments
Indicates current step number if program is in operation

Save Restore

The Micromax Set Point Program Configuration display shows all the steps in a Set Point Program. You can edit these steps or create new steps if you wish.

Editing Steps in a Micromax set point program

To edit steps (or create new steps) in a Set Point Program, you must be logged on to Station with a security level of Supervisor (or higher).

To edit a step of the Micromax Set Point Program change the appropriate fields as necessary:

- Type
- Start Value (in EU) or Soak Value (in EU)
- Rate (EU/min) or Soak Time (min)
- Events

Creating new steps

To create a new step, change the step **Type** of the first available blank step to either **Ramp** or **Soak**.

Steps must be created in the correct order. You cannot create a step if the preceding steps have not yet been configured. You can create a maximum of 32 steps for a Micromax Set Point Program.

Deleting steps

To delete a step, set the step **Type** to blank.



Note

Deleting a step will delete all subsequent steps in the Program.

Restoring configuration settings

To discard the changes that you have made to a Set Point Program in a current editing session, click the **Restore** button on the Set Point Program Configuration display. This restores the Program's configuration settings that were last saved to the LPU using the Save button.



Note

If you leave a Set Point Program Configuration display without saving the changes, any changes you have made will be lost.

Saving changes

To save the changes that you have made in a Set Point Program to the Micromax LPU, click the **Save** button on the Set Point Program Configuration display (Figure 7).

Creating a new set point program

To create a new Set Point Program:

- 1 Make sure you are logged on to Station at SUPR security level (or higher).
- 2 Select **Configure > Applications ⇒ Micromax ⇒ Set Point Programs**.
This calls up the Micromax LPU Set Point Program Summary display (“Micromax Set point program summary display” on page 38).
- 3 Click the **Programmer Number** of the programmer for which the Set Point Program will be created. The Micromax LPU Set Point Program details are shown for that programmer (see “Set Point Program Configuration Display” on page 39).
- 4 Enter step data for the new Set Point Program.
If a Program is already configured, clear the current configuration by select for Step 1 a step Type of **blank**.
- 5 To write the new Set Point Program to the controller, click **Save**.



Note

If you leave a Set Point Program Configuration display without saving the changes, any changes you have made will be lost.

Trending Micromax set point programs

Micromax Set Point Programs can be viewed on a Trend display, showing the Set Point Program plotted against the PV of the loop that the Set Point Program is driving.

Before the Set Point program can be trended:

- 1 Configure a point to scan the process variable (PV) that has its set point driven by the Set Point Program.
- 2 Additionally this point must have its A1 and A2 parameters configured to scan the status of the Set Point Program (SPPSTS x U32B where x is the Set Point program number) and the current step number of the Set Point Program (SPPS x IEEEEFP where x is the Set Point Program number).
- 3 The PV, A1, and A2 parameters must also have history assigned.
If the point has been configured with history on its A1 and A2 parameters, the Set Point Programs profile will be plotted against the PV of the point entered over the last run of the Set Point Program.
If history has not been configured or not enough history is available, only the Set Point Programs profile will be plotted.

For information about creating a point to address a Micromax LPU and scan its data, see “Defining a Micromax address for a point parameter value” on page 20.

To view a Set Point Program Trend:

- 1 Call up the Trends display (select **Trends** from the System Menu display).
- 2 Click an unused trend.
- 3 For **Trend Set**, select **UMAX SPP**.
- 4 In the **Point ID to Trend** field, enter the ID of a point that has a PV scanning the appropriate loop output that you want to view.



Note

There can be a mismatch of up to 1 history interval period between the process variable being plotted (blue trace) and the Set Point Program's profile (yellow trace). The history interval period is shown in the top left of the display.

Configuring Micromax processes

Processes are created to define locations on the Micromax network where recipes can be downloaded to or uploaded from. Processes can be viewed and modified from the Micromax LPU Processes display on Station (Figure 8). To call up this display select **Configure** ⇒ **Applications** > **Micromax** ⇒ **Recipe Management** ⇒ **Processes**.

Figure 8 LPU Process Summary Display

System Configuration

General

System Hardware

Areas

Alarm & Event Management

Operator Security

History

Reports

Schedules

Trend & Group Displays

Acronyms

Applications

Application Development

Server Scripting

Micromax LPU Processes

Process Number	Name	Description	Controller	Programmers	Constants
1	Furnace	Main production furnace	UMAX1	2	26 - 50
2				None	None
3				None	None
4				None	None
5				None	None
6				None	None
7				None	None
8				None	None
9				None	None
10				None	None

Recipes

Upload/Download

The Micromax LPU Processes display shows:

- The process names.
- A description of the process.
- The associated controller.
- The programmer as programmers in the controller to be associated with the process. (Programmers are the part of the LPU which run SPPs.)
- The constant blocks in the LPU to be used.

Adding or changing a process

Existing processes are changed by selecting different values from the lists for **Controller**, **Programmers**, and **Constants**.
To add a process, find an empty location, enter a **Name** and **Description** for the process and then define the process by selecting values from the lists for **Controller**, **Programmers**, and **Constants**.

Deleting a process

A process is deleted by clearing the **Name** and the **Description** of the process in the respective fields, and setting the configuration values of the process (all three lists) to blank or none.

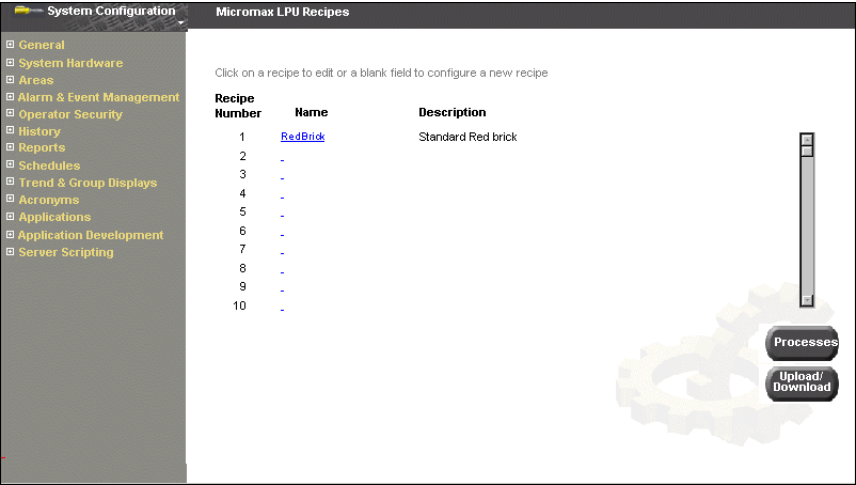
Configuring Micromax recipes

To:	Go to:
View a Recipe summary	page 45
Modify a Recipe configuration	page 46
Upload and download Recipes	page 47

Viewing the Micromax recipe summary

Micromax Recipes can contain 0, 1, or 2 Set Point Programs and 0, 25, 50, or 100 constants. Recipes are viewed and modified on the Micromax LPU Recipes display on Station (“Micromax LPU recipes summary display” on page 45). To call up this display select **Configure > Applications ⇒ Micromax ⇒ Recipe Management ⇒ Recipes**.

Figure 9 Micromax LPU recipes summary display



Viewing and modifying a Micromax recipe configuration

To view or modify a recipe, click the recipe **Name** on the Micromax LPU Recipes display. This calls up the Micromax LPU Recipe configuration display for that recipe (see “Recipe configuration display” on page 46).

Figure 10 Recipe configuration display

The following items can be edited from the Micromax Recipe configuration display:

- Recipe name
- Recipe description
- Whether the recipe’s access to processes is restricted to processes that exactly match the recipe
- What processes the recipe is associated with
- What the recipe contains—the Set Point Program (SPP) count (0, 1 or 2 SPPs), and the Constants (CN) count (0 to 4 for lots of 25, that is 0, 25, 50, or 100 constants)

Viewing or modifying the recipe set point programs and constants

The recipe contents can be viewed or modified using the Edit buttons, **A**, **B**, **C**, and **D** in the Recipe Contents section of the display. If the recipe is configured with 1 or 2 SPPs, the appropriate Edit button calls up the Micromax LPU Recipe SPP display for that SPP. Clicking an **Edit** button that is not configured for the recipe has no effect. The same procedure applies to the editing of constants.

Clicking the appropriate **Edit** button calls up the Micromax LPU Recipe Constants configuration display.

A recipe must be associated with at least one process to be uploaded or downloaded. See “Uploading and downloading recipes” on page 47. The process specifies which controller, program(s) and constant block(s) will be used by the recipe.

Modifying Micromax recipe set point program configuration

Recipe Set Point Programs are modified on the Micromax LPU Recipe SPP configuration display. The procedures used are the same as those for editing the Micromax LPU Set Point Programs described in “Viewing or editing set point programs” on page 39.

Modifying the Micromax recipe constants configuration

Recipe Constants are modified on the Micromax LPU Recipe Constants configuration display. To edit Constants, enter a new number in the **Setting** field for the item.

Uploading and downloading recipes

The Micromax LPU Recipe Upload/Download display is used to upload recipes from the LPU or download them to the LPU. To reach this display, either select **Configure > Applications > Micromax > Recipe Management > Upload/Download** from the Station menu, or click the **Upload/Download** button on the Micromax LPU Recipes display or Micromax LPU Processes display.

Before a recipe is uploaded or downloaded the following needs to be done:

- Configure an appropriate process that defines the location of the recipe in a controller. This step can be skipped and a process built on the Upload/Download display.
- Configure a recipe to contain the information which is being uploaded or downloaded. This recipe should have the process configured above listed in its associated processes.

To upload or download a recipe:

- 1 From the **Recipe** list, select the appropriate recipe.
- 2 Select the process which the recipe is either being uploaded from or downloaded to.
 - To select an existing process, select **Use Existing Process** and select the process from the **Process** list. The process description displays.
 - If an appropriate process has not already been set up, you can configure one by selecting **Use selected Controller/Programmers/Constants** and then making your selections from the respective lists, Controller, Programmer(s), Constants.
- 3 Click the **Upload** or **Download** buttons as appropriate. All selections on this display will be either uploaded or downloaded respectively.

Index

A

A-NAK messages, in response to read or write requests 37

B

baud rates 11

Black Box converter, connecting with 9

C

cabling, for Micromax connection to the server 9

channel properties, for Micromax 14

Configure

 Micromax Recipes 45

connection to the server

 serial connection options 9

controller properties, for Micromax 18

D

deleting

 Micromax processes 44

G

generic channels

 Port tab 15

M

Micromax

 I/O address mapping 22

 value mapping to the server 23

Micromax LPU 38

Micromax process

 deleting 44

Micromax Processes

 configure 43

Micromax Recipe CN Configuration 47

Micromax Recipes 45, 46

Micromax recipes

 Upload/Download 47

Micromax Set Point Programs 38–48

multi-dropping 9

N

networking Micromax LPUs 8

P

parity 11

Port tab 15

Property Page

 Port tab 15

S

Stallion EasyConnection adapter, connecting with 10

system architecture, examples 8

T

terminating resistor 11

terminating resistor 11

termination, of RS232/422 communications links 9

U

unit address 11

Upload/Download

 Micromax recipes 47

Honeywell RM7800 Flame Safeguard Interface Reference

Issue

6.1

Date

November 2005

Notice

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 Limited Australia.

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.

Copyright 2005 – Honeywell Limited Australia

Honeywell trademarks

PlantScape[®], SafeBrowse[®], **TotalPlant**[®] and TDC 3000[®] are U.S. registered trademarks of Honeywell International Inc.

Experion[™] and Honeywell Enterprise Buildings Integrator[™] are trademarks 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.

Support and other contacts

For technical assistance, call your nearest Honeywell office.

Contents

1	Getting started	5
	Support and documentation for 7800 Series	6
	7800 Series models supported	6
	Other documentation for 7800 Series	6
	7800 Flame Safeguard-specific terms	7
2	7800 Series controller setup	9
	Connecting 7800 Series to the server	10
	Using an RS-232/485 converter	11
	Using an RS-485 adapter	13
	Communications with the server	15
	Obtaining subnetwork numbers for multi-dropped devices	15
	Communication settings	15
	Tuning QS7850A GPI performance parameters	17
3	7800 Flame Safeguard configuration and addressing	19
	Defining a 7800 Flame Safeguard channel	20
	7800 Flame Safeguard channel main properties	20
	7800 Flame Safeguard channel port properties	22
	Defining a 7800 Flame Safeguard controller	24
	7800 Flame Safeguard controller main properties	24
	Defining a 7800 Flame Safeguard address for a point parameter value	27
	Entering an address	27
	Address and data format definitions	27
	Points addressable from the server in the RM78XX and EC78XX range	28
	Optimizing scanning performance	32
	7800 Flame Safeguard scan packets	32
4	Server and Station tasks for 7800 Flame Safeguard	33
	Testing server to 7800 Flame Safeguard communications	34
	Accessing 7800 Series data for troubleshooting	35
	7800 SERIES Alarm Text reference	36

Getting started

1

This reference describes how to set up, configure, and test 7800 Flame Safeguard controller communications with the server. There is detailed information for defining the controller in Quick Builder. Complete each step before starting the next. The order of configuration should not be changed.

Steps for connecting and configuring a 7800 Flame Safeguard controller:

Steps	Go to:
Set Up Q7700 Network Interface Unit and connect each 7800 Series controller to the Network Interface Unit	page 10
Connect the Q7700 Network Interface Unit to the server via a QS7850 General Purpose Interface Module	page 10
Configure the QS7850 module using a terminal emulation package	page 15
Define channels in the Quick Builder project	page 20
Define controllers in the Quick Builder project	page 24
Download channel and controller definitions to the server	
Test communications	page 34
Use Quick Builder to configure points	page 27

Support and documentation for 7800 Series

7800 Series models supported

The 7800 Series Burner Control Interface is designed for and tested with the following 7800 Series Devices.

- **Q7700 Network Interface Unit**
Requires the QS7850 GPI module installed for communication with the server
- **QS7850 General Purpose Interface module**
Software revision 1.5 or later
- **RM78XX Burner Controller**
Must be connected to a QS7800A or QS7800B Module in the Q7700
- **EC78XX Burner Controller**
Must be connected to a QS7800A or QS7800B Module in the Q7700
- **QS7800A Communication Interface ControlBus Module**
For direct connection to a single Burner controller
- **QS7800B Communication Interface ControlBus Module**
Connects multi-dropped Burner controllers by way of S7810B switch unit
- **S7810B Switch Unit**
Required on each Burner controller multi-dropped from a QS7800B

Other documentation for 7800 Series

The documents listed are relevant to the configuration of 7800 Series devices and should be read prior to the installation of the interface. Refer also to the documentation specific to the model of Relay Module that you are using.

- Q7700A,B Network Interface Unit
- SYSNet™ QS7800A Network Interface ControlBus™ Module for 7800 SERIES Relay Modules
- QS7850A General Purpose Interface ControlBus™ Module
- Combustion System Manager (CSM) User's Guide

7800 Flame Safeguard-specific terms

Network Interface Unit (NIU)

A Q7700 with a General Purpose Interface (QS7850A) card installed. Flame Safeguard Controllers are attached to this.

General Purpose Interface (GPI) Module

A module that fits into a slot on the Q7700 to enable communication with other devices over a RS-485 connection.

7800 Series controller setup

2

This chapter describes how to set up a 7800 Flame Safeguard controller for the system.

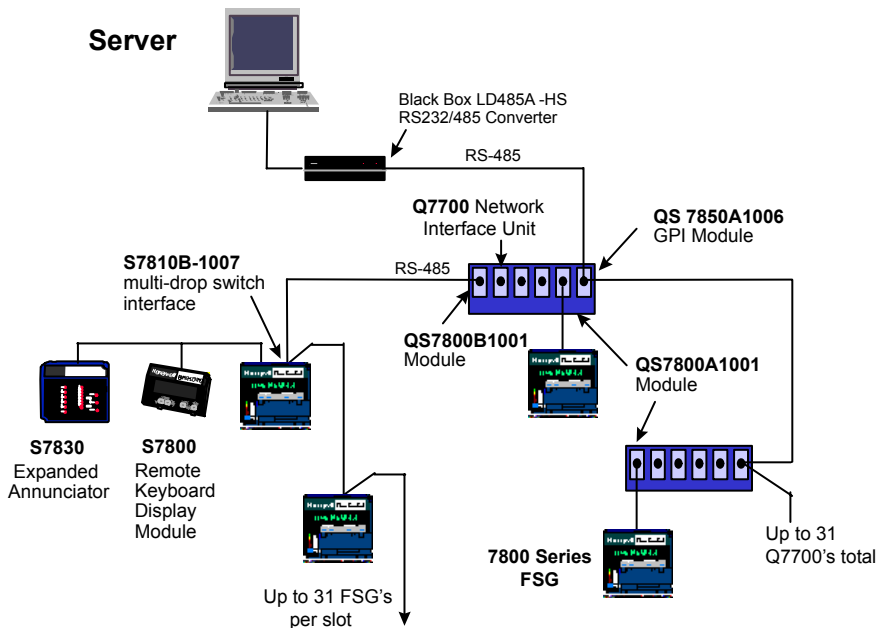
The tasks for setting up a 7800 Series controller are:

For	Go to:
Connecting to the server	page 10
Communication settings	page 15
Tuning QS7850A GPI performance parameters	page 15

Connecting 7800 Series to the server

- Connect each 7800 Series device to the Q7700 Network Interface Unit using a QS7800A, or QS7800B, Communication Interface ControlBus Module.
- Connect the Q7700 to the server using a QS7850 GPI module.
- 7800 Series devices support connection to the server via RS-485. This allows either one or more devices to be connected to the server.

Figure 1 Server connection to the 7800 Series controllers



It is possible to multidrop QS7850 GPI modules and hence Q7700s on the same RS-485 link. It is also possible to multidrop 7800 Series controllers on the same ControlBus line connected to the one QS7800B Module. In this case, each controller must have a dedicated Switch Unit, S7810B. For more details see *Q7700A,B Network Interface Unit* documentation. The configuration of this network must then involve the Combustion System Management (CSM) software.

For performance reasons, restrict the number of 7800 relays per channel further than its theoretical maximum. Typically, no more than 32 per channel should be configured. Less if greater performance is required. For details, see “7800 Flame Safeguard scan packets” on page 32.

Using an RS-232/485 converter

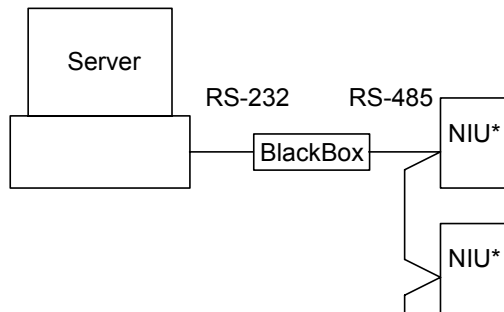
Honeywell recommends that you use the Black Box LD485-HS RS-232 to RS-485 Interface Converter. This converter has been qualified by Honeywell. Use of another converter may produce unexpected results.



Note

Be sure to order the correct model number of this converter for the voltage in your location.

Figure 2 Black Box Converter



Connect the devices as shown in “Wiring diagram for connection to Black Box Converter” on page 12:

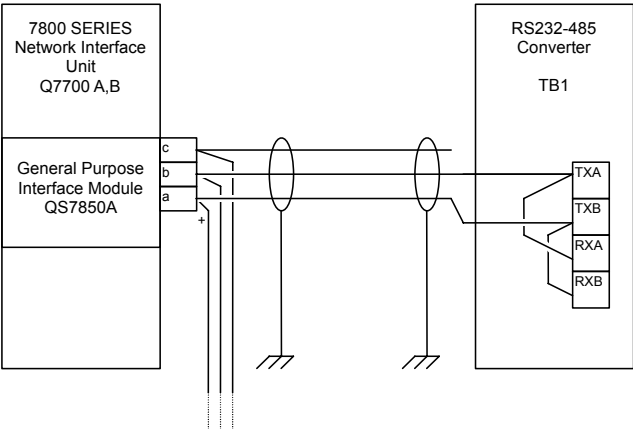
- Connect the Network Interface Units (NIUs) on the RS-485 network to the converter.
- Connect the RS-232 port of the Black Box converter to an RS-232 port on the server using a straight through cable.

The following settings should be made on the Black Box Converter (ME837A).

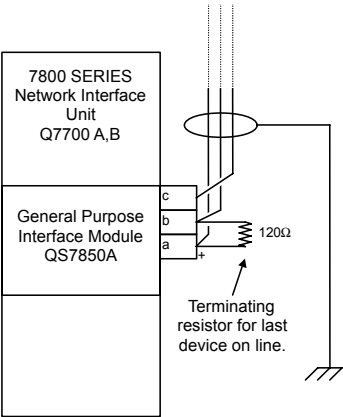
Switch	Setting
XW1A	Jumper in
XW1B	Jumper out
W8	B-C
W9	D
W15	BC
W17	C

Switch	Setting
W16	B
S1	Jumper out
S2	On
S3	On

Figure 3 Wiring diagram for connection to Black Box Converter



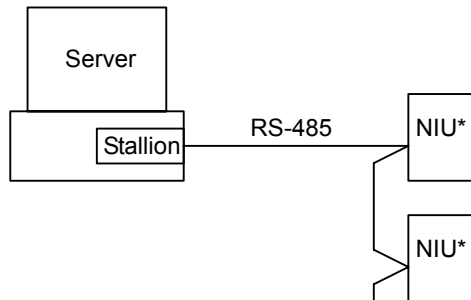
(to other 7800 SERIES
Network Interface Units
with General Purpose
Interface Modules)



Using an RS-485 adapter

Honeywell recommends using the Stallion EasyConnection 8/32 adapter, part number EC8/32-AT-NT-K with the RS-232 to RS-485 8 port module, part number PA-EC-8D4-K (Rev 2). This adapter has been qualified by Honeywell. Use of another adapter may produce unexpected results.

Figure 4 Stallion RS-485 adapter connection

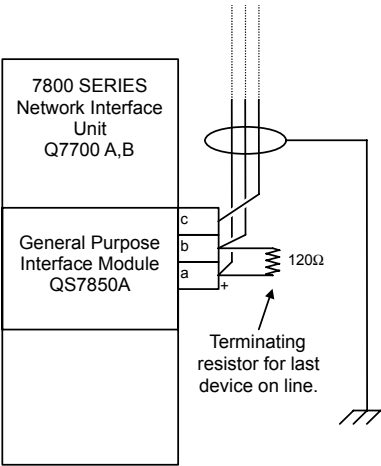
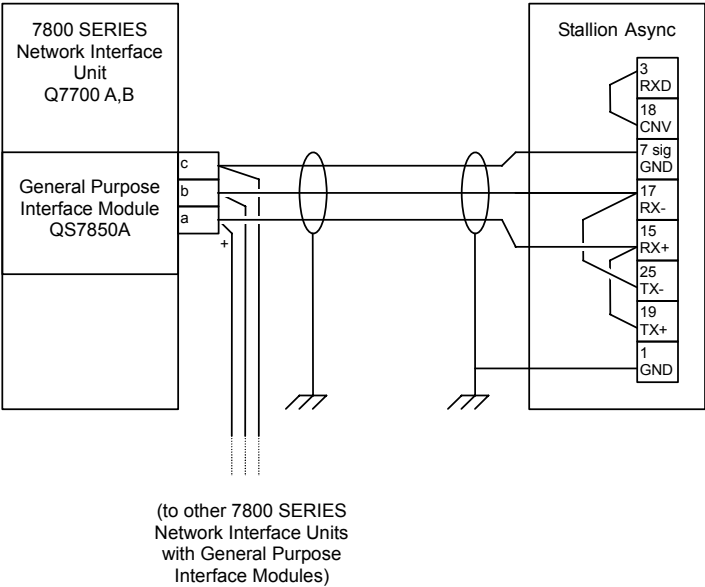


To connect:

- 1 Install the adapter and driver in the server as described in the adapters documentation.
- 2 Connect a port on the RS-485 adapter (installed on the server) directly to the RS-485 network to which the Network Interface Units (NIUs) are connected. This is a 2-wire RS-485 connection.

For additional information consult the Stallion documentation.

Figure 5 Wiring diagram for Stallion adapter



Communications with the server

Most of the communication parameters are not configurable.

Obtaining subnetwork numbers for multi-dropped devices

When devices are multi-dropped from one QS7800B, configuration of the multi-drop network must be done by Combustion System Management (CSM) software. CSM assigns subnetwork numbers to the multi-drop busses that are connected via the S7810Bs.

To find out what the subnetwork numbers are, you need to print a site configuration. To print, select the site icon in CSM's very first screen, then select **File > Print**.

The printout shows the subnetwork numbers used by an S7810B adjacent to the serial number of the S7810B so that you can match up each one of these numbers with the physical S7810B that uses it.

Communication settings

When a single NIU and GPI are attached to the RS-485 bus, a node ID of 0 can be used to address the GPI card.



Note

All GPI cards respond to being addressed by a node ID of 0 regardless what their actual address is.

When NIUs are multi-dropped on the same RS-485 bus, the node ID of each GPI card must be set to a unique number in the range of 8 to 254.

To set the node ID:

- 1 Use Terminal Emulation software to connect to the serial port on which the GPI is attached (via an RS-232 to RS-485 Interface Converter). Use these settings:
BAUD Rate = 9600
Data Bits = 8
Stop bits = 1
Parity = None
- 2 Ensure there is only one GPI card on the line at this time.

- 3 Type **ss** to run the system status command to show the current status of the card.

If the following response is not obtained, check your cabling and communication settings and try again.

The response should be similar to the following example.

Example

```
Alarm      = 0
Comm       = 41
Default    = 0 0
EOL        = crlf
Binary     = On
ASCII      = On
Mapping    = On
Node       = 0
Pacing (0.1s) = 5
Queue Size = 20
Reply Timeout (s) = 6
ASCII Timeout (s) = 30
Binary Timeout (s) = 1
Turnaround (ms) = 6
Listen Interval (ms) = 40
Zapped     = N
ROM Version = 1.6.
```

- 4 When this response is obtained, run the command **sn VV** where *vv* is the unique node ID you wish to set for this GPI card.
The response from the GPI card for this command should be “ok”.
- 5 Confirm the change has taken effect by performing another **ss** command and verifying the node ID returned.

See also “Tuning QS7850A GPI performance parameters” on page 17.

Tuning QS7850A GPI performance parameters

There are two other tuning parameters within the GPI that should be modified for better performance. They are “Pacing” and “Timing” and are set from the Terminal Emulation software.

Tuning parameter	Default	Recommended
Pacing	5 (0.5 seconds per message)	1 (0.1 seconds per message)
Timing	6 seconds	12 seconds

To set pacing and timing:

- 1 Type **sp 1** to set the pacing.
The message “ok” appears to indicate that the pacing has been set.
- 2 Type **st 12** to set the timing to 12 seconds.
The message “ok” appears to indicate that the timing has been set.
- 3 Type **ss** to confirm the changes have taken effect.
Output similar to the example should appear.

Example

```

Alarm      = 0
Comm       = 41
Default    = 0 0
EOL        = crlf
Binary     = on
ASCII      = on
Mapping    = on
Node       = 0
Pacing (0.1s) = 1
Queue Size = 20
Reply Timeout (s) = 12
ASCII Timeout (s) = 30
Binary Timeout (s) = 1
Turnaround (ms) = 6
Listen Interval (ms) = 40

Zapped = N
ROM Version = 1.6

```


7800 Flame Safeguard configuration and addressing

3

This chapter describes how to configure a 7800 Flame Safeguard controller for the system using Quick Builder. For each configuration procedure, there is detailed information covering all supported 7800 Series devices.

The Quick Builder controller configuration tasks are described:

For:	Go to:
Steps for defining a 7800 Flame Safeguard channel	page 20
Steps for defining a 7800 Flame Safeguard controller	page 24
Defining an address for a point parameter value	page 27
How to optimize scanning performance	page 32

Defining a 7800 Flame Safeguard channel

You need to define a channel for the serial connection the server uses to communicate with the 7800 Flame Safeguard controllers.

To define a channel using Quick Builder:

- 1 Click  to add a channel.
- 2 In the Add Items dialog box, select `channel` as the item and `7800 Series` as the channel type. Complete the number of items and name details and click **OK**.
- 3 Enter the channel item details on the Main property page for the 7800 Flame Safeguard channel. For help with the channel definitions, see 7800 Flame Safeguard channel main properties.
- 4 To complete the channel definition, click the Port tab and define either a serial or a terminal server port. See “7800 Flame Safeguard channel port properties” on page 22.

7800 Flame Safeguard channel main properties

Use this property page to name a 7800 Flame Safeguard channel and define its properties. A channel definition is required by the server for the communications link.

Property	Description
Name	The unique name of the channel. A maximum of 10 alphanumeric characters (no spaces or double quotes).
Description	(Optional) Type a description of the channel. A maximum of 30 characters can be used, including spaces.

Property	Description
Marginal Alarm Limit	The communications alarm marginal limit at which the channel is declared to be marginal. When this limit is reached, a high priority alarm is generated. A channel barometer monitors the total number of requests and the number of times the controller did not respond or response was incorrect. The barometer increments by 2 or more, depending on the error and decrements for each good call. To calculate an acceptable limit, multiply the square root of the number of controllers on the channel by the Marginal Alarm Limit defined for those controllers. (Normally, you specify the same value for all controllers on a channel). For example, if there are 9 controllers on the channel and their Marginal Alarm Limit is set to 25, the value would be $[\sqrt{9} \times 25 = 75]$.
Fail Alarm Limit	Communications alarm fail limit at which the channel is declared to have failed. When this barometer limit is reached, an urgent alarm is generated. Set this to double the value specified for the channel Marginal Alarm Limit.
Diagnostic Scan Rate	The rate (in seconds) at which the server polls failed controllers attached to this channel to detect when communication is re-established. The default is 60 seconds.
Connect Timeout	Amount of time (in seconds) the server waits to connect to the controller before abandoning the connection. The default is 10 seconds. Use the default value unless the communications line has a high error rate or unless you are using modems.
Read Timeout	Amount of time (in seconds) the server waits for a reply from the controller. Type a new value in this field if you do not want the default, 2 seconds. Use the default value unless the communications line has a high error rate or unless you are using modems.
Item Type	Shows the type of item specified when this item was created.
Last Modified	Shows the date of the most recent modification to this channel's property details.
Last Downloaded	This shows the date that the item was last downloaded to the server.

Property	Description
Item Number	The unique item number currently assigned to this item. You can change the item number displayed in this field if you need to match your current server database configuration. The item number must be between 1 and the maximum number of channels allowed for your system.

7800 Flame Safeguard channel port properties

Enter the port description for the 7800 Flame Safeguard channel on the Port property page. The port definition must match your connection and can be either:

- “Serial port properties” on page 22
- “TerminalServer port properties” on page 23

Serial port properties



Note

The serial port settings must match the settings on your communication devices.

Property	Description
Serial Port Name	The device name of the serial port.
Baud Rate	The number of data bits per second. The rate must be set to 9600 .
Number of Data Bits	The number of data bits used for transmission. The number must be 8 .
Stop Bits	The number of stop bits used for transmission. The number must be 1 .
Parity	The parity verification used on the port. Parity must be NONE .
Checksum	Not applicable for 7800 Series.
XON/XOFF	Set to None (the default).
RS-232	Not applicable for 7800 Series. Do not use.

Property	Description
RS-485	Only applicable for a an RS-485 link. (The RS-485 and RS-232 settings are mutually exclusive.) Enable Stallion RS-485 Half Duplex. Select this if you are using a Stallion EasyConnection. Don't select this if you use a 4-wire Stallion RS-422 connection. Echo. Select this to indicate that the server should expect the messages it sends to the port on the transmit line to be echoed back on the receive line. For example, select this if you are using a Stallion EasyConnection adapter. For RS-485 connections using a Black Box Converter, leave the box unselected (no echo).

TerminalServer port properties

Use a terminal server to connect controllers with a serial interface to a LAN. Use this port type to define your terminal server port connection for the server channel.

Property	Description
Terminal Server TCP Host Name	The name and port number of terminal server to which the channel is connected.
Terminal Server TCP Port No	You can specify either a TCP host name or an IP address, but it must match the TCP host name used when you installed and internally configured the terminal server.
Idle Timeout	The time, in seconds, the channel waits for a successful connection to the server before closing the connection. A value of 0 indicates that the connection is never closed.
Checksum	Not applicable for a 7800 Series controller.

Defining a 7800 Flame Safeguard controller

Define a controller for each Flame Safeguard controller connected to the server via the Q7700.

To define a controller using Quick Builder:

- 1 Click  to add a controller.
- 2 In the Add dialog box, select **Controller** as the item and **7800 Flame Safeguard** as the controller type. Complete the number of items and name details and click **OK**.
- 3 Enter the property definitions for the controller on the controller Main property page. See “7800 Flame Safeguard controller main properties” on page 24.
- 4 Download your controller and channel definitions to the server and test the communications. See “Testing server to 7800 Flame Safeguard communications” on page 34.

7800 Flame Safeguard controller main properties

Use this page to define the 7800 Flame Safeguard controller.

Property	Description
Name	The unique name of the controller. A maximum of 10 alphanumeric characters (no spaces or double quotes). For LAN connected controllers, the name must not contain underscore (_) characters. This name is used to look up the IP address in the TCP/IP database if you do not specify an IP Address property.
Description	(Optional) Type a meaningful description for this controller. A maximum of 30 characters, including spaces, can be entered.
Channel Name	The name of the channel on which the controller communicates. In the list of channel names, click the name. You need to have defined the channel in order for it's name to appear in the list.

Property	Description
Marginal Alarm Limit	The communications alarm marginal limit at which the controller is declared to be marginal. When this value is reached, a high priority alarm is generated. This limit applies to the controller barometer which monitors the total number of requests to the controller and the number of times the controller did not respond or response was incorrect. The barometer increments by 2 or more, depending on the error and decrements for each good call. The default value is 25.
Fail Alarm Limit	The communications alarm fail limit at which the controller is declared to have failed. When this barometer value is reached, an urgent alarm is generated. Set this to double the value specified for the controller Marginal Alarm Limit .
Controller Type	From the list, select the controller type. One of either: • EC78XX • RM78XX
SubNet Number (1 to 254)	Depending on the connection, enter either a Q7700 unit slot number, or a subnetwork number: • For a single Burner controller attached to the QS7800A module in the Q7700 Unit, specify the slot number in which this module is inserted (1 to 6). • For one of several Burner controllers attached to the same QS7800B module (multi-dropped), enter the subnetwork number. Use the Combustion System Management (CSM) software to obtain the subnetwork number. See “Obtaining subnetwork numbers for multi-dropped devices” on page 15.
Network ID	Enter the node number (0 to 254) of the QS7850 General Purpose Interface module (GPI) in the Q7700 unit to which the device is attached. If there is only one QS7850 attached to the channel, enter 0 .
Item Type	Shows the type of item specified when this item was created.
Last Modified	Shows the date of the most recent modification to this channel’s property details.
Last Downloaded	This shows the date that the item was last downloaded to the server.

Property	Description
Item Number	The unique item number currently assigned to this item. You can change the item number displayed in this field if you need to match your current server database configuration. The number must be between 1 and the maximum number of controllers allowed for your system.

Defining a 7800 Flame Safeguard address for a point parameter value

Entering an address

For source and destination addresses, the format for a 7800 Flame Safeguard controller address is:

ControllerName Location

Part	Description
<i>ControllerName</i>	The name of the 7800 Flame Safeguard controller.
<i>Location</i>	The location in the controller where the value is recorded. Specify the mnemonic displayed in “Points addressable from the server in the RM78XX and EC78XX range” on page 28.

If you would like help when defining an address, click  next to **Address** to display Address Builder. For details, see the help.

Address and data format definitions

The points that are available from the server will vary depending on the particular device connected.

There are no points within the Q7700 that can be referenced from server points.

Address

The location should specify the mnemonic displayed in “Points addressable from the server in the RM78XX and EC78XX range” on page 28.

Data formats

The data format for a point is automatically determined according to the address mnemonic used, however, only points with the following data types can be read from the controller.

- 1-byte unsigned integer
- 2-byte unsigned integer
- 2-byte signed fixed-point
- 4-byte signed integer

Points addressable from the server in the RM78XX and EC78XX range

RM78XX Point	EC78XX Point	Number	Description
PRM_MAJSWREV	PEC_MAJSWREV	1	Major_SW_Rev
PRM_MINSWREV	PEC_MINSWREV	2	Minor_SW_Rev
PRM_SWVER	PEC_SWVER	3	SW_Version
PRM_SAVEX	PEC_SAVEX	5	Saved_Index_X
PRM_MSGFG	PEC_MSGFG	6	Message_Flags
PRM_RWSQSTAT	PEC_RWSQSTAT	7	Raw_Seq_State
PRM_XACOD	PEC_XACOD	8	Raw_xacod
PRM_AVAIL	PEC_AVAIL	9	BurnerAvailability
PRM_REMO	PEC_REMO	10	Remote_Command
PRM_FLAMS	PEC_FLAMS	11	Flame_Signal
PRM_INABT0	PEC_INABT0	12	Run/Test_Switch
PRM_INABT1	PEC_INABT1	13	PilotValveHold_T16
PRM_INABT2	PEC_INABT2	14	Manual-Open_Sw_T17
PRM_INABT3	PEC_INABT3	15	LowFire_Switch_T18
PRM_INABT4	PEC_INABT4	16	HighFireSwitch_T19
PRM_INABT5	PEC_INABT5	17	Valve_2/Start_T21
PRM_INABT6	PEC_INABT6	18	PreIgn_ILK_T20
PRM_INBBT1	PEC_INBBT1	19	Oper_Control_T6
PRM_INBBT2	PEC_INBBT2	20	Ignition_T10
PRM_INBBT3	PEC_INBBT3	21	Main_Valve_T9
PRM_INBBT4	PEC_INBBT4	22	Pilot_Valve_T8
PRM_INBBT6	PEC_INBBT6	23	Interlock_T7
PRM_RWRLY	PEC_RWRLY	24	Relay_Command
PRM_PURG	PEC_PURG	25	Purge_Time
PRM_AMP	PEC_AMP	26	Amplifier_Type
PRM_JMPR1	PEC_JMPR1	27	Jumper_1
PRM_JMPR2	PEC_JMPR2	28	Jumper_2
PRM_JMPR3	PEC_JMPR3	29	Jumper_3
PRM_FFRT	PEC_FFRT	30	Flame_Response
PRM_FLAME	PEC_FLAME	31	Flame_Status
PRM_SAVEY	PEC_SAVEY	34	Saved_Index_Y
PRM_FLTIX	PEC_FLTIX	35	Fault_Index

RM78XX Point	EC78XX Point	Number	Description
PRM_ACCLO	PEC_ACCLO	36	Acc_Lockout
PRM_SQ	PEC_SQ	37	Sequence_Status
PRM_SQEX	PEC_SQEX	38	Sequence_Extension
PRM_SQTM	PEC_SQTM	39	Sequence_Timer
PRM_CYC	PEC_CYC	40	Total_Cycles
PRM_HRS	PEC_HRS	41	Total_Hours
PRM_FC1	PEC_FC1	42	H1_Fault_Code
PRM_SFM1	PEC_SFM1	43	H1_Fault_Message
PRM_SQ1	PEC_SQ1	44	H1_Sequence_Status
PRM_SQEX1	PEC_SQEX1	45	H1_Sequence_Ext.
PRM_SQTM1	PEC_SQTM1	46	H1_Sequence_Timer
PRM_CYC1	PEC_CYC1	47	H1_Fault_Cycle
PRM_HRS1	PEC_HRS1	48	H1_Fault_Hour
PRM_FC2	PEC_FC2	49	H2_Fault_Code
PRM_SFM2	PEC_SFM2	50	H2_Fault_Message
PRM_SQ2	PEC_SQ2	51	H2_Sequence_Status
PRM_SQEX2	PEC_SQEX2	52	H2_Sequence_Ext.
PRM_SQTM2	PEC_SQTM2	53	H2_Sequence_Timer
PRM_CYC2	PEC_CYC2	54	H2_Fault_Cycle
PRM_HRS2	PEC_HRS2	55	H2_Fault_Hour
PRM_FC3	PEC_FC3	56	H3_Fault_Code
PRM_SFM3	PEC_SFM3	57	H3_Fault_Message
PRM_SQ3	PEC_SQ3	58	H3_Sequence_Status
PRM_SQEX3	PEC_SQEX3	59	H3_Sequence_Ext.
PRM_SQTM3	PEC_SQTM3	60	H3_Sequence_Timer
PRM_CYC3	PEC_CYC3	61	H3_Fault_Cycle
PRM_HRS3	PEC_HRS3	62	H3_Fault_Hour
PRM_FC4	PEC_FC4	63	H4_Fault_Code
PRM_SFM4	PEC_SFM4	64	H4_Fault_Message
PRM_SQ4	PEC_SQ4	65	H4_Sequence_Status
PRM_SQEX4	PEC_SQEX4	66	H4_Sequence_Ext.
PRM_SQTM4	PEC_SQTM4	67	H4_Sequence_Timer
PRM_CYC4	PEC_CYC4	68	H4_Fault_Cycle
PRM_HRS4	PEC_HRS4	69	H4_Fault_Hour

RM78XX Point	EC78XX Point	Number	Description
PRM_FC5	PEC_FC5	70	H5_Fault_Code
PRM_SFM5	PEC_SFM5	71	H5_Fault_Message
PRM_SQ5	PEC_SQ5	72	H5_Sequence_Status
PRM_SQEX5	PEC_SQEX5	73	H5_Sequence_Ext.
PRM_SQTM5	PEC_SQTM5	74	H5_Sequence_Timer
PRM_CYC5	PEC_CYC5	75	H5_Fault_Cycle
PRM_HRS5	PEC_HRS5	76	H5_Fault_Hour
PRM_FC6	PEC_FC6	77	H6_Fault_Code
PRM_SFM6	PEC_SFM6	78	H6_Fault_Message
PRM_SQ6	PEC_SQ6	79	H6_Sequence_Status
PRM_SQEX6	PEC_SQEX6	80	H6_Sequence_Ext.
PRM_SQTM6	PEC_SQTM6	81	H6_Sequence_Timer
PRM_CYC6	PEC_CYC6	82	H6_Fault_Cycle
PRM_HRS6	PEC_HRS6	83	H6_Fault_Hour
PRM_RESETRTL	PEC_RESETRTL	84	Reset_Trail
PRM_DIAGNO	PEC_DIAGNO	85	Diagnostics
PRM_WARNING	PEC_WARNING	86	Warning
PRM_HZ60	PEC_HZ60	87	Line_Frequency
PRM_ISGENDIAG	PEC_ISGENDIAG	88	Generic_Label
PRM_REVDEVTYPE	PEC_REVDEVTYPE	89	Reverse_EC/RM
PRM_REVFFRT	PEC_REVFFRT	90	Reverse_FFRT
PRM_SAVY	PEC_SAVY	91	Save_Y
PRM_COMPAT	PEC_COMPAT	92	Compatibility
PRM_START	PEC_START	93	Start_Code
PRM_INABT7	PEC_INABT7	94	CS_Asic
PRM_RWINA	PEC_RWINA	95	Raw_Port_A
PRM_INBBT0	PEC_INBBT0	96	FlockFeedback
PRM_INBBT5	PEC_INBBT5	97	SafetyRly_FB
PRM_INBBT7	PEC_INBBT7	98	Reset
PRM_RWINB	PEC_RWINB	99	Raw_Port_B
PRM_RLYBT0	PEC_RLYBT0	100	RlyBit0
PRM_RLYBT1	PEC_RLYBT1	101	RlyBit1
PRM_RLYBT2	PEC_RLYBT2	102	RlyBit2
PRM_RLYBT3	PEC_RLYBT3	103	RlyBit3

RM78XX Point	EC78XX Point	Number	Description
PRM_RLYBT4	PEC_RLYBT4	104	RlyBit4
PRM_RLYBT5	PEC_RLYBT5	105	RlyBit5
PRM_RLYBT6	PEC_RLYBT6	106	RlyBit6
PRM_RLYBT7	PEC_RLYBT7	107	RlyBit7
PRM_RWJMPR	PEC_RWJMPR	108	Raw_Jumper
PRM_FRATE	PEC_FRATE	109	Fire_Rate_Status

Optimizing scanning performance

The maximum amount of data that can be acquired from a controller is influenced by the rate of sending scan packets to the controller. An understanding of the 7800 Series scan packets will help you configure points so that optimal data acquisition performance can be achieved by maximizing the amount of data acquired with each scan packet.

The scan packets that have been built can be listed by using the utility **lisscn** (list scan). Listing scan packets helps verify the scanning strategy. See the *Configuration Guide* for your server for usage of **lisscn**.

7800 Flame Safeguard scan packets

7800 Flame Safeguard Scan packets can contain up to 14 points of any of the supported data types from the one controller. The GPI card in a Network Interface Unit is relatively slow to respond to requests so it is important to minimize the load on the GPI. Depending on the exact architecture of the network, the maximum throughput is approximately 1.5 packets per second.

Scan packets result from:

- Periodic scanning of configured points. Point parameters are scanned in groups of up to 14 addresses maximum per packet. To be in the same scan packet the points need to have the same scan period and reside in the same controller, otherwise multiple scan packets will result.
- Diagnostic display updates. An update of the Diagnostic display requires approximately four scan packets. The display is updated both when it is first called up and when the refresh button is pressed.

For each channel, the total load from the above sources must be assessed to ensure that it does not exceed the maximum load. For example, if you configure a network with five controllers and wish to scan a single point, the flame signal, from each controller at a five-second update rate, the calculation is:

$$\frac{(5 \text{ controllers}) * (1 \text{ packet each})}{5 \text{ second scan rate}} = 1 \text{ packet per second}$$

This is less than 1.5 and therefore acceptable.

Server and Station tasks for 7800 Flame Safeguard

4

This chapter describes tasks for the 7800 Flame Safeguard controller that you perform either on the server or from any Station.

For:	Go to:
Testing communications with the server	page 34
Accessing 7800 Series data for troubleshooting	page 35
Viewing the current state of the controller	page 35
Alarm text reference for 7800 Series	page 36

Testing server to 7800 Flame Safeguard communications

Use the interface test utility, **h7800tst**, to test communications between the server and the connected devices after you have downloaded your channel and controller definitions to the server database.



Note

There are no communications with the Q7700 Network Interface Unit itself, therefore you need to have a complete working link to at least one Burner controller before testing can proceed.

Before testing make sure you have completed the following:

- Set up the 7800 controllers.
- Connected all cables.
- Entered definitions for controller and channel in Quick Builder project file.
- Downloaded project file definitions to server without errors.

Considerations

- The server need not be running while using the utility as long as the database service is running.

To run the test utility:

- 1 Click **Start > Programs > Honeywell Experion PKS> Server > Diagnostic Tools > 7800 Series Burner Control Test Utility**.
- 2 Follow the directions as prompted.
- 3 If the installation and configuration of the controller has been successful you will be able to use the menu options to read a point in the controller. It will be difficult to verify the value of the point, but any successful read will indicate the channel and controller configuration is complete.

If any errors are encountered, review the Quick Builder project details for the 7800 Series channel and controller.

When communications with the controller have been verified it is possible to view all the details that are visible on the Controller Diagnostic display when Station is running and the channel has been enabled. See the *Configuration Guide* for your server for information on enabling channels and calling up the Controller Diagnostic display.

Accessing 7800 Series data for troubleshooting

7800 Series data is accessible from Station for a particular device after the controller has been configured and successfully downloaded.

- Diagnostic data—available on the Controller Diagnostic display on Station as summary data. This information is available even before any points have been configured and downloaded to the server. Use the Controller Diagnostic display on Station to view the current state of the controller and to analyze why the burner tripped.
- Alarm data—available on the Alarm Summary display on Station after one or more points have been built in the controller. The fault index of the controller generates the alarms.

For information on how to navigate to these displays, see the *Configuration Guide*.

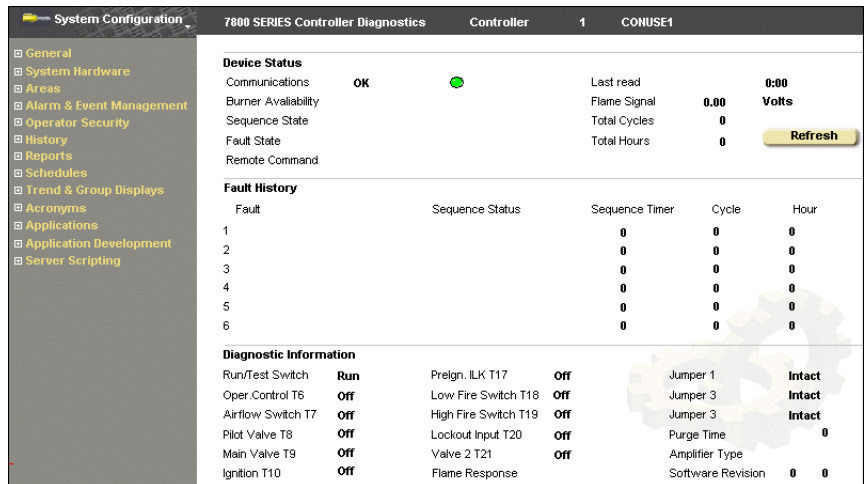
To access any other values from the device, or to use any of the values shown on the Controller Diagnostic display, points must be configured.



Note

If communications with the controller are not successful, the message “Configuration Unavailable” is displayed and the Controller Diagnostic display does not appear.

Figure 6 7800 Series Controller Diagnostic display



7800 SERIES Alarm Text reference

No Fault
Flt 1: No Purge Card
Flt 2: AC Frequency/Noise
Flt 3: AC Line Dropout
Flt 4: AC Frequency
Flt 5: Low Line Voltage
Flt 6: Purge Card Error
Flt 7: Flame Amplifier
Flt 8: Flame Amp/Shutter
Flt 9: Flame Detected
Flt 10: PreIgn. Interlock
Flt 11: Running Interlock On
Flt 12: Lockout Interlock On
Flt 13: Airflow Switch On
Flt 14: High Fire Switch
Flt 15: Flame Detected
Flt 16: Flame-Out Timer
Flt 17: Main Flame Failure
Flt 18: Flame Detected
Flt 19: Main Flame Ignition
Flt 20: Low Fire Switch Off
Flt 21: Running Interlock
Flt 22: Lockout Interlock
Flt 23: Airflow Switch
Flt 24: Call Service
Flt 25: Call Service
Flt 26: Manual-Open Sw. Off
Flt 27: Start Switch On
Flt 28: Pilot Flame Failure
Flt 29: Lockout Interlock
Flt 30: Running Interlock
Flt 31: Low Fire Switch Off

Flt 32: Airflow Switch
Flt 33: PreIgn. Interlock
Flt 34: Oper. Control On
Flt 35: Call Service
Flt 36: Call Service
Flt 37: Call Service
Flt 38: Call Service
Flt 39: Call Service
Flt 40: Call Service
Flt 41: Main Valve On
Flt 42: Pilot Valve On
Flt 43: Ignition On
Flt 44: Pilot Valve 2 On
Flt 45: Low Fire Switch Off
Flt 46: Flame Amp Type
Flt 47: Jumpers Changed
Flt 48: Delayed MV On
Flt 49: Manual-Open Sw. On
Flt 50: Jumpers Wrong
Flt 51: Flame Too Strong
Flt 52: Call Service
Flt 53: Lockout Switch
Flt 54: Comb. Pressure
Flt 55: Purge Fan Sw On
Flt 56: Block Intake
Flt 57: Purge Fan Sw Off
Flt 58: Call Service
Flt 59: Call Service
Flt 60: Call Service
Flt 61: Call Service
Flt 62: Call Service
Flt 63: Call Service
Flt 64: Call Service
Flt 65: Call Service
Flt 66: Call Service

Flt 67: AC Phase
Flt 68: Preignition ILK
Flt 69: Call Service
Flt 70: Call Service
Flt 71: Device Specific
Flt 72: Device Specific
Flt 73: Device Specific
Flt 74: Device Specific
Flt 75: Device Specific
Flt 76: Accessory Flt
Flt 77: Accessory Flt
Flt 78: Accessory Flt
Flt 79: Accessory Flt
Flt 80: Accessory Flt
Flt 81: Accessory Flt
Flt 82: Accessory Flt
Flt 83: Accessory Flt
Flt 84: Accessory Flt
Flt 85: Accessory Flt
Flt 86: Accessory Flt
Flt 87: Accessory Flt
Flt 88: Accessory Flt
Flt 89: Accessory Flt
Flt 90: Accessory Flt
Flt 91: Accessory Flt
Flt 92: Accessory Flt
Flt 93: Accessory Flt
Flt 94: Call Service
Flt 95: Call Service
Flt 96: Call Service
Flt 97: Call Service
Flt 98: Call Service
Flt 99: Call Service
Flt 100: Call Service
Flt 101: Call Service

Flt 102: Call Service
Flt 103: Call Service
Flt 104: Call Service
Flt 105: Call Service
Flt 106: Call Service
Flt 107: Call Service
Flt 108: Call Service
Flt 109: Call Service
Flt 110: Call Service
Flt 111: Call Service
Flt 112: Call Service
Flt 113: Call Service
Flt 114: Call Service
Flt 115: Call Service
Flt 116: Call Service
Flt 117: Call Service
Flt 118: Call Service
Flt 119: Call Service
Flt 120: Call Service
Flt 121: Call Service
Flt 122: Call Service
Flt 123: Call Service
Flt 124: Call Service
Flt 125: Call Service
Flt 126: Call Service
Flt 127: Call Service
Flt 128: Poor Flame Sensor
Flt 10y: Valve Closure
Flt 10z: Other PreIgn. ILK
Flt 21a: Aux. Limit #1
Flt 21b: Aux. Limit #2
Flt 21c: LWCO
Flt 21d: High Limit
Flt 21e: Aux. Limit #3
Flt 21f: Fuel Select Off

Flt 21g: Both Fuels Selected

Flt 21h: High Oil Pressure

Flt 21i: Low Oil Pressure

Flt 21j: High Oil Temp.

Flt 21k: Low Oil Temp.

Flt 21m: Atomizing Sw.

Flt 21n: High Gas Pressure

Flt 21o: Low Gas Pressure

Flt 21p: Airflow Switch

Flt 21q: Aux. Interlock #4

Flt 21r: Aux. Interlock #5

Flt 21s: Other Interlocks

Flt 22a: Aux. Limit #1

Flt 22b: Aux. Limit #2

Flt 22c: LWCO

Flt 22d: High Limit

Flt 22e: Aux. Limit #3

Flt 22f: Fuel Select Off

Flt 22g: Both Fuels Selected

Flt 22h: High Oil Pressure

Flt 22i: Low Oil Pressure

Flt 22j: High Oil Temp.

Flt 22k: Low Oil Temp.

Flt 22m: Atomizing Sw.

Flt 22n: High Gas Pressure

Flt 22o: Low Gas Pressure

Flt 22p: Airflow Switch

Flt 22q: Aux. Interlock #4

Flt 22r: Aux. Interlock #5

Flt 22s: Other Interlocks

Flt 23a: Aux. Limit #1

Flt 23b: Aux. Limit #2

Flt 23c: LWCO

Flt 23d: High Limit

Flt 23e: Aux. Limit #3

Flt 23f: Fuel Select Off
Flt 23g: Both Fuels Selected
Flt 23h: High Oil Pressure
Flt 23i: Low Oil Pressure
Flt 23j: High Oil Temp.
Flt 23k: Low Oil Temp.
Flt 23m: Atomizing Sw.
Flt 23n: High Gas Pressure
Flt 23o: Low Gas Pressure
Flt 23p: Airflow Switch
Flt 23q: Aux. Interlock #4
Flt 23r: Aux. Interlock #5
Flt 23s: Other Interlocks
Flt 29a: Aux. Limit #1
Flt 29b: Aux. Limit #2
Flt 29c: LWCO
Flt 29d: High Limit
Flt 29e: Aux. Limit #3
Flt 29f: Fuel Select Off
Flt 29g: Both Fuels Selected
Flt 29h: High Oil Pressure
Flt 29i: Low Oil Pressure
Flt 29j: High Oil Temp.
Flt 29k: Low Oil Temp.
Flt 29m: Atomizing Sw.
Flt 29n: High Gas Pressure
Flt 29o: Low Gas Pressure
Flt 29p: Airflow Switch
Flt 29q: Aux. Interlock #4
Flt 29r: Aux. Interlock #5
Flt 29s: Other Interlocks
Flt 30a: Aux. Limit #1
Flt 30b: Aux. Limit #2
Flt 30c: LWCO
Flt 30d: High Limit

Flt 30e: Aux. Limit #3
Flt 30f: Fuel Select Off
Flt 30g: Both Fuels Selected
Flt 30h: High Oil Pressure
Flt 30i: Low Oil Pressure
Flt 30j: High Oil Temp.
Flt 30k: Low Oil Temp.
Flt 30m: Atomizing Sw.
Flt 30n: High Gas Pressure
Flt 30o: Low Gas Pressure
Flt 30p: Airflow Switch
Flt 30q: Aux. Interlock #4
Flt 30r: Aux. Interlock #5
Flt 30s: Other Interlocks
Flt 32a: Aux. Limit #1
Flt 32b: Aux. Limit #2
Flt 32c: LWCO
Flt 32d: High Limit
Flt 32e: Aux. Limit #3
Flt 32f: Fuel Select Off
Flt 32g: Both Fuels Selected
Flt 32h: High Oil Pressure
Flt 32i: Low Oil Pressure
Flt 32j: High Oil Temp.
Flt 32k: Low Oil Temp.
Flt 32m: Atomizing Sw.
Flt 32n: High Gas Pressure
Flt 32o: Low Gas Pressure
Flt 32p: Airflow Switch
Flt 32q: Aux. Interlock #4
Flt 32r: Aux. Interlock #5
Flt 32s: Other Interlocks
Flt 33y: Valve Closure
Flt 33z: Other PreIgn. ILK

Index

Numerics

7800 Series
 data access 35
 device requirements and support 6
 other documentation 6

A

addressing, 7800 Series controller 27
architecture 10

C

channel definition, for 7800 Series 20
communication settings for 7800 Series 15
communication, testing 34
configuring 7800 Series, steps for 5
controller
 addressing 7800 Series 27
 definition for 7800 Series 24

D

data access on Station 35
data formats 27
documentation, for 7800 Series 6

N

Network Interface Unit, definition 7
node ID, for multi-drop NIUs 15

P

pacing, setting 17
points, addressable in the RM78XX and EC78XX
 range 28
port definition, for 7800 Series 22

R

RS-232/485 converter
 architecture 11
 recommended 11
RS-485 adapter
 recommended 13

S

scan packets, load for 7800 Series 32
Stallion EasyConnection adapter, connecting 13
subnetwork numbers, for multi-dropped
 devices 15

T

timing, setting 17
tuning QS7850A GPI performance
 parameters 17

Honeywell Series 9000 Integration Reference

Issue

9.1

Date

November 2005

Notice

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 Limited Australia.

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.

Copyright 2005 – Honeywell Limited Australia

Honeywell trademarks

PlantScape[®], SafeBrowse[®], **TotalPlant**[®] and TDC 3000[®] are U.S. registered trademarks of Honeywell International Inc.

Experion[™] and Honeywell Enterprise Buildings Integrator[™] are trademarks 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.

Support and other contacts

For technical assistance, call your nearest Honeywell office.

Contents

1	Getting started	5
	Support and documentation for Series 9000	6
	Series 9000 models supported	6
	Other documentation for Series 9000	6
	Series 9000-specific terms	7
2	Series 9000 controller setup	9
	Setting the ELPM IP address	10
	Connecting Series 9000 to the server	11
	Ethernet connection to Series 9000 using ELPM	11
	Choosing a Series 9000 configuration method	12
3	Series 9000 controller configuration and addressing	13
	Defining a Series 9000 channel	14
	Series 9000 channel main properties	14
	Series 9000 port properties	15
	Defining a Series 9000 controller	16
	Series 9000 controller Main properties	16
	Creating points in Quick Builder after using Series 9000 Configuration Software	18
	Defining a Series 9000 address for a point parameter value in Quick Builder	19
	Entering an address	19
	Using Address Builder to enter an address	19
	Scaling with data formats	24
	Optimizing scanning performance	26
	Series 9000 scan packets	26
4	Server and Station tasks for Series 9000	29
	Testing server to Series 9000 communications	30
	Troubleshooting communications errors	32
	Viewing and modifying Series 9000 block parameters	33
	Configuring Series 9000 set point programs and recipes	34
	Configuring Series 9000 set point programs (SPP)	34
	Viewing Series 9000 set point program trends	39
	Configuring Series 9000 recipes	40

Getting started

1

This reference describes how to set up, configure, and test Series 9000 controller communications with the server. If you are using the Series 9000 Configuration Software, you will need to read the *Series 9000 Configuration Software Manual*.

Complete each step before commencing the next.

Steps for Connecting and Configuring a Series 9000 Controller.

Steps	Go to:
Set the IP Address in the ELPM	page 10
Connect the cables from the server to the Series 9000 controller	page 11
Select a Series 9000 configuration method	page 12
Define channels:	
• With Quick Builder	page 14
• With Configuration Software	
Define controllers	
• With Quick Builder	page 16
• With Configuration Software,	
Download channel and controller definitions to the server	
Test communications	page 30
Troubleshooting communication errors	page 32

Support and documentation for Series 9000

Series 9000 models supported

The server supports the following Series 9000 controller models:

- 9000e
- 9100e
- 9200e

Other documentation for Series 9000

- *Series 9000 Communications Reference Manual* 71-SE-18-01
- *Series 9000 Loop and Logic Block Configuration* 51-51-10-16

Series 9000-specific terms

It is assumed that the reader is familiar with the components, operation and configuration of the Series 9000 equipment. A clear understanding of these terms is necessary for accurate configuration of the server database and fault diagnosis. You should refer to the documentation listed in “Support and documentation for Series 9000” on page 6 for complete information on the Series 9000.

ELPM

Ethernet Loop Processor Module. The ELPM is part of the Series 9000. It provides the Ethernet connection to the Series 9000. It also performs the loop processing functions of the Series 9000.

LPM

Loop Processor Module

Series 9000 controller setup

2

This chapter describes how to set up a Series 9000 controller for the system.

There are two ways to configure a Series 9000 controller, using the Series 9000 Configuration Software or using Quick Builder. Make sure you read and understand the restrictions that apply if you want to use Quick Builder to configure points after you use the Configuration Software.

The tasks for setting up a Series 9000 controller are:

For	Go to:
Setting the ELPM IP address	page 10
Connecting to the server	page 11
Choosing a configuration method	page 12

Setting the ELPM IP address

Set the IP address using the DIP switches on the ELPM installed in the Series 9000. The IP address must be unique for all devices on the LAN.

Example	ELPM DIP switches S5 set to Internet address 200.0.0.16			
	bit	76543210	76543210	76543210
	off(=0)	00110111	11111111	11101111

For more information about installing the ELPM, see the *S9000 Controller Installation Manual*.

To the server, each Series 9000 controller is treated as three controllers:

- One for PLC data (logic processor data)
- One for LPM control block configuration
- One for LPM control block I/O

The IP address you set for the Series 9000 must be defined for each controller you configure in the server. There are three different ways to do this:

- Use the Series 9000 Configuration software to create controllers. The Configuration Software adds the necessary entries to the hosts file.
- Define the IP address for a controller on the Main property page for the controller in Quick Builder. See also “Defining a Series 9000 controller” on page 16.
- Define the IP address for a controller in the TCP/IP database of the server.

The server uses the controller name assigned in Quick Builder to look up the IP address. As each controller uses the same IP address, when adding the address to the TCP/IP database of the server, use the controller names as aliases so that the server can determine their IP address.

Example	TCP/IP host file entry for a Series 9000 controller: 200.0.0.16 plcio cbcfg cbio
	where 200.0.0.16 is the IP address and “plcio”, “cbcfg”, and “cbio” correspond to the names of the three server controllers used to access the different types of data in the Series 9000.

Connecting Series 9000 to the server

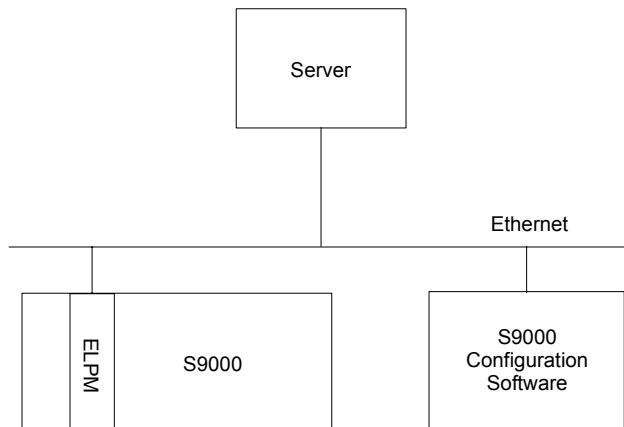
Series 9000 is connected to the server by way of thin Ethernet LAN or Cat 5 twisted pair and an AUI transceiver.

Communication is by way of the ELPM.

Ethernet connection to Series 9000 using ELPM

An Ethernet connection to an Series 9000 using ELPM.

Figure 1 Ethernet connection to Series 9000 using ELPM



Note

- The ELPM does not support IP routing.
 - Release 200.3 of the Series 9000 Configuration Software is only supported on Windows 95 and Windows NT 4.
-

Choosing a Series 9000 configuration method

There are two ways in which you can configure connections and points in the server for Series 9000 controllers. The method you select will depend on whether or not you want to make use of database integration features.

Considerations

- During an upgrade from a previous release you are required to upload your converted server database to Quick Builder, regardless of whether you used server connected mode or Quick Builder to configure controllers.

If you make changes after the upgrade to the Series 9000 configuration using the Server Connected mode, these changes are not included in Quick Builder until you upload the Quick Builder database.

Alternatively, if you are not going to configure the Series 9000 using Quick Builder you can delete the Series 9000 channels, controllers and points from Quick Builder.

Method	Description
Server connected mode. Use the Series 9000 Controller Configuration Software R200.3.	Provides server and controller database integration. As well as configuring the controller, it can also configure the server database for the Series 9000 including channel, controller, and point definitions.
Server Unconnected mode. Use Quick Builder.	Can be used to configure the server when not using the integrated database features of the Series 9000 Configuration Software. Quick Builder can be used to configure Series 9000 channel, controller, and point definitions in the server database. You must use Quick Builder if you want to enhance point functionality with algorithms or if there are other controller types.



Note

The Series 9000 Configuration Software is needed for the initial setup and programming of the controller. It is not needed for server configuration in unconnected mode.

Series 9000 controller configuration and addressing

3

This chapter describes how to configure a Series 9000 controller for the system using Quick Builder. For each configuration procedure, there is detailed information covering all supported Series 9000 devices.

The Quick Builder controller configuration tasks are described:

For:	Go to:
Steps for defining a Series 9000 channel	page 14
Steps for defining a Series 9000 controller	page 16
Defining an address for a point parameter value	page 19
Address syntax for PLC Register addressing	page 19
Address syntax for LPM I/O addressing	page 22
Creating points in Quick Builder after using Configuration Software	page 18
How to optimize scanning performance	page 26

Defining a Series 9000 channel

You need to define a channel for the LAN connection the server uses to communicate with the Series 9000 controllers.

To define a channel using Quick Builder:

- 1 Click  to add a channel.
- 2 In the Add Items dialog box, select channel as the item and s9000 as the channel type.
- 3 Enter the channel item details on the Main property page for the Series 9000 channel. For help with the channel definitions, see Series 9000 channel main properties.
- 4 To complete the channel definition, click the Port tab and define a LAN port. See “Series 9000 port properties” on page 15.

Series 9000 channel main properties

Use the Main tab to enter basic properties for the channel.

Property	Description
Name	The unique name of the channel. A maximum of 10 alphanumeric characters (no spaces or double quotes).
Description	(Optional) Type a description of the channel. A maximum of 30 characters can be used, including spaces.
Marginal Alarm Limit	The communications alarm marginal limit at which the channel is declared to be marginal. When this limit is reached, a high priority alarm is generated. A channel barometer monitors the total number of requests and the number of times the controller did not respond or response was incorrect. The barometer increments by 2 or more, depending on the error and decrements for each good call. To calculate an acceptable limit, multiply the square root of the number of controllers on the channel by the Marginal Alarm Limit defined for those controllers. (Normally, you specify the same value for all controllers on a channel). For example, if there are 9 controllers on the channel and their Marginal Alarm Limit is set to 25, the value would be [3 is square root] x 25= 75.

Property	Description
Fail Alarm Limit	Communications alarm fail limit at which the channel is declared to have failed. When this barometer limit is reached, an urgent alarm is generated. Set this to double the value specified for the channel Marginal Alarm Limit.
Connect Timeout	Amount of time (in seconds) the server waits to connect to the controller before abandoning the connection. Type a new value in this field if you do not want the default, 10 seconds. Use the default value unless the communications line has a high error rate or unless you are using modems.
Read Timeout	Amount of time (in seconds) that the server waits for a reply from the controller. Type a new value in this field if you do not want the default, 2 seconds. Use the default value unless the communications line has a high error rate or unless you are using modems.
Item Type	Shows the type of item specified when this item was created.
Last Modified	Shows the date of the most recent modification to this channel's property details.
Last Downloaded	This shows the date that the item was last downloaded to the server.
Item Number	This field displays the unique item number currently assigned to this item by Quick Builder. You can change the item number displayed in this field if you need to match your current server database configuration. The item number must be between 1 and the maximum number of channels allowed for your system.

Series 9000 port properties

The Port tab defines the communication-related properties for a channel. The only **Port Type** you can define for Series 9000 is LANVendor.

LANVendor port properties

Enter the following descriptions for the Series 9000 channel:

Property	Description
Port Name	Leave this field blank.

Defining a Series 9000 controller

Each Series 9000 connected with ELPM requires a different controller definition to access each one of the three data tables.

To define a controller using Quick Builder:

- 1 Click  to add a controller.
- 2 In the Add Items dialog box, select **Controller** as the item and **Series 9000** as the controller type.
- 3 Enter the property definitions for the controller on the controller Main property page.

Series 9000 controller Main properties

Use the Main tab to define the basic properties for a Series 9000 controller.

Property	Description
Name	The unique name of the controller. A maximum of 10 alphanumeric characters (no spaces or double quotes). The name must not contain underscore (_) characters. This name is used to look up the IP address in the TCP/IP database if you do not specify an IP Address property.
Description	(Optional) Type a meaningful description for this controller. A maximum of 30 characters, including spaces, can be entered.
Channel Name	The name of the channel on which the controller communicates. In the list of channel names, click the name. You need to have defined the channel in order for its name to appear in the list.
Marginal Alarm Limit	The communications alarm marginal limit at which the controller is declared to be marginal. When this value is reached, a high priority alarm is generated. This limit applies to the controller barometer which monitors the total number of requests to the controller and the number of times the controller did not respond or response was incorrect. The barometer increments by 2 or more, depending on the error and decrements for each good call. The default value is 25.

Property	Description
Fail Alarm Limit	The communications alarm fail limit at which the controller is declared to have failed. When this value is reached, an urgent alarm is generated. Set this to double the value specified for the controller Marginal Alarm Limit.
IP Address	(Optional) The IP address must match that set for the Series 9000 ELPM. Do not enter the IP address here if you have entered it in the server TCP/IP database. For details, see “Setting the ELPM IP address” on page 10.
Data Table	Select one of: • LPM Control Block I/O • PLC Registers & Digitals • LPM Control Block Config
Item Type	Shows the type of item specified when this item was created.
Last Modified	Shows the date of the most recent modification to this channel’s property details.
Last Downloaded	This shows the date that the item was last downloaded to the server.
Item Number	This field displays the unique item number currently assigned to this item by Quick Builder. You can change the item number displayed in this field if you need to match your current server database configuration. The number must be between 1 and the maximum number of controllers allowed for your system.

Creating points in Quick Builder after using Series 9000 Configuration Software

Quick Builder can be used to build additional points for functions not supported by the Series 9000 Configuration Software, for example algorithms.



Caution

If points have been built in server connected mode, you will need to update your Quick Builder database.

If additional points need to be built, perform the following steps:

- 1 Start a new project in Quick Builder.
- 2 Select **Tool > Upload**.
- 3 When the upload is completed successfully, further configuration can be performed.

Defining a Series 9000 address for a point parameter value in Quick Builder

Entering an address

For source and destination addresses the format for a Series 9000 controller address is:

ControllerName Location

Part	Description
<i>ControllerName</i>	The name of the Series 9000 controller.
<i>Location</i>	The location in the controller where the value is recorded. The syntax can be entered as either: • “PLC register and digital addressing” on page 19 • “LPM I/O and configuration data addressing” on page 22

If you would like help when defining an address, click  next to **Address** to display Address Builder. For details, see the help.

Using Address Builder to enter an address

PLC register and digital addressing

Address [Format | BitNumber]

Part	Description
<i>Address</i>	The server addressable memory address within the PLC data table. For details, see page 20.

Part	Description
<i>Format</i>	(Optional) Select an appropriate format to read raw values: 0 = Reads 16-bit integers from 0 to 65536. IEEEFP = Reads IEEE floating point numbers. IEEEFP numbers use two data addresses, address and (address-1). Notes: • Be careful not to overlap two IEEEFP data addresses. • Do not set an IEEEFP address to 4096.
<i>BitNumber</i>	(Optional) For parameter values with addresses greater than 4095, enter the starting bit number. The valid range is 0 (default) to 15, where 0 is the least significant bit. The number of bits read is implied by the point type. Analog point parameters read up to 16 bits; status point parameters read 1, 2, or 3 consecutive bits.

Example**For analog point**

PV Source: 5051 IEEEFP

SP Source: 5054 U999

SP Destination: 5054 U999

For accumulator point

PV Source: 5070

For status point

PV Source: 322 (bit 0)

PV Source: 5001 3 (register 5001, bit 3)

Point parameter source/destination address ranges

The address range varies according to the table type.

**Note**

Some models of Series 9000 have smaller address ranges than those listed. Consult the Series 9000 documentation for the precise address range of your particular model.

Table Type	Address Range	Point Parameter		
		Status	Analog	Accum
Real DI/DO	0 to 2047	PV, OP, MD	MD	
Internal DI/DO	2048 to 4095	PV, OP, MD	MD	
Register	4096 to 8191	PV, OP, MD	PV, OP, SP, MD, A1 to A4	PV

Address restrictions

Real analog Input/Output (0-2047) is not directly addressable.

Use relay ladder logic to move the analog value to and from a register.

Analog input cards are 12 bit; use data format U4095.

System status information, including redundancy status, is not directly addressable.

Use relay ladder logic to move this to internal I/O or registers for access.

Series 9000 memory maps vary according to the model. See the Series 9000 documentation for details.

Data format for scaling

Data formats for scaling are valid for analog points only. The default format is **U4095**.

Data Format	Counts
U4095	0 to 4095
U9999	0 to 9999
U999	0 to 999

Data format for reading floating point values

Select format **IEEEFP** to read two consecutive registers as a single precision floating point number.



Note

- IEEEFP numbers use two data addresses, address and address-1. Do not specify address-1 (the lower address) as the point parameter location address.
- When configuring controller data tables, do not assign overlapping data addresses if floating point values are used and do not set an IEEEFP address to any value less than 4097.

Data format for reading raw values without scaling

Enter **0** to read all 16 bits in the register without scaling. To read less than 16 bits without scaling, enter the starting bit number (1 to 15).

LPM I/O and configuration data addressing

The location syntax for an LPM I/O address is:

ControlBlockNumber [*ControlBlockType*] *ParameterNumber* [*Format*]

Part	Description
<i>ControlBlockNumber</i>	A number from 1 to 250. To access operator tags, use block number 251 on an LPM I/O type controller (data table type 2).
<i>ControlBlockType</i>	(Optional) The LPM block type. There is no default. A block number needs to be associated with a block type in order to access the Series 9000 Block Configuration pages.
<i>ParameterNumber</i>	Within the LPM control block, a number from 0 to 31. Parameter numbers for each block type can be found in the <i>S9000 Communications Reference Manual</i> . Different block types can use a different number for the same parameter. For example, the parameter number for the Input PV parameter of the PID1 block is 11 but for the PID3 block it is 15. Data table type 1 has configuration parameters; data table type 2 has input/output parameters. If accessing operator tag information, the parameter number is from 0 to 149. Note: If the data format is MODE, the parameter number is ignored.

Part	Description
<i>Format</i>	(Optional) To read the raw values, select the appropriate format. • IEEEFP = Reads IEEE floating point data (the default). • INT4 = Reads 32-bit integer data (not usually used). • MODE = PID type block mode parameter. When this format is used, the parameter number is ignored to enable composite mode support in PID blocks. For details, see “Composite mode support in PID blocks” on page 24. • REVERSE = Reverses a digital value (0 becomes 1, 1 becomes 0).

**Note**

There are still some PID parameters that require the controller to go into PROGRAM mode before a change can be made, for example the PID type A or B. These configuration parameters must be set by using the Series 9000 Configuration software.

Example**For analog point**

PV Source, 5/PID2 15

MD Source, 5/PID2 31 MODE

OP Source, 5 13 (no block type)

For status point

PV Source, 251 88 (operator tag example)

Operator tag support

To access information in an operator tag, use block number 251 and an index number between 0 and 149.

The index number corresponding to an operator tag is found in the Series 9000 Configuration Software Continuous Control chart.

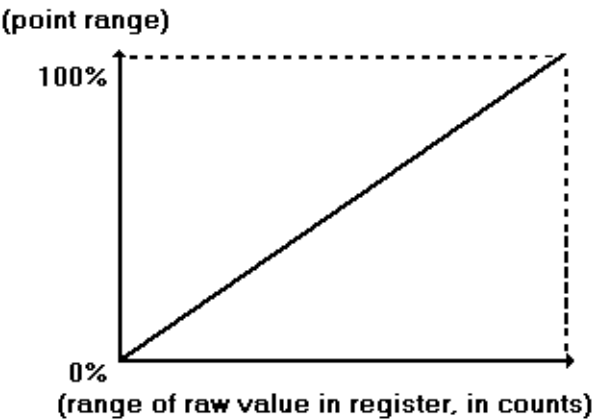
Scaling with data formats

Point parameter values that have addresses in the registers can be scaled with a data format. Select the format that corresponds to the counts that have been set in the PLC register.

Data Format	Counts in PLC Register
U4095 (default)	0 to 4095
U9999	0 to 9999
U999	0 to 999

The data format tells the server how to interpret the register value. Raw values in the PLC register tables can be scaled by the 0% and 100% point range values.

To select a format for scaling, select the format that corresponds to the counts that have been set in the register where the point parameter value is sourced.



Composite mode support in PID blocks

The mode (MD) point parameter can be used to monitor and control both the remote setpoint and the manual modes of a PID block.

When monitoring and controlling, the server reads and writes both the PID block manual output mode request (parameter number 2) and remote setpoint mode request (parameter number 3) I/O parameters.

To enter the controller address for the mode point that will read and write to both control block parameter addresses, enter:

- PID Control Block number
- Any parameter number
- MODE for the data format

Whichever I/O parameter number you enter, it is ignored if the data format is MODE.

The server and the Series 9000 use different acronyms for the same modes.

Server Mode Acronym	Series 9000 PID Block Mode
MAN	LSP-MAN or RSP-MAN
AUTO	LSP-AUTO
CASC	RSP-AUTO

These acronyms are stored in the server system acronym table and can be changed either at installation time or after installation using a Station display. See the *Configuration Guide* for information about modifying acronyms.

Optimizing scanning performance

The maximum amount of data that can be acquired from an controller is influenced by the rate of sending scan packets to the controller. An understanding of the Series 9000 scan packets will help you configure points so that optimal data acquisition performance can be achieved by maximizing the amount of data acquired with each scan packet. For a discussion of scan packets, see the *Configuration Guide*.

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

Series 9000 scan packets

For PLC digital memory, the values for up to 1920 consecutive addresses can be acquired in one scan packet.

Figure 2 Scan packet format for digital values

Starting Address (a multiple of 16)
Number of Addresses (maximum of 1920)

For PLC Registers and LPM data, the values for up to 60 scattered addresses can be acquired in one scan packet.

Figure 3 Scan packet format for PLC registers or LPM values

Address 1
Address 2
...
...
...
...
Address 60

Two classes of scan packet are used for Series 9000:

- Hardware Diagnostic Scan
- Periodic Data Acquisition Scan

Server and Station tasks for Series 9000

4

This chapter describes tasks for the Series 9000 controller that you perform either on the server or from any Station.

For:	Go to:
Testing communications with the server	page 30
Troubleshooting communication errors	page 32
Viewing and modifying block parameters	page 33
Configuring set point programs and recipes	page 34

Testing server to Series 9000 communications

A Series 9000 diagnostic utility, **s90tst**, is included as part of the server software. You can test communications between the server and the Series 9000 after channel and controller definitions are downloaded to the server.

Before testing:

- Set up Series 9000
- Connect cables
- Download channel and controller definitions without errors

If there is a Loader Terminal with the system, it should be used to ensure that the Series 9000 is operating correctly.



Note

The utility can be used with the server either on- or off-line.

To test communications:

- 1 In a Command Prompt window, type **s90tst** and press ENTER.
- 2 Follow the screen prompts and type and enter the values relevant to the data you are testing.

To display help when running **s90tst**, Type **?** and press ENTER.

Example test of PLC data

This example reads a status (digital) value from address 1 of the logic processor.

Enter LRN or device name of channel: chn07

Enter the controller number (item number in Quick Builder): 50

SERIES 9000 PLC Input/Output

Enter function to be performed

- 0 - QUIT S90TST communications utility
- 1 - READ data
- 2 - WRITE data
- 3 - SELECT a new controller

Enter function: 1

Status or Registers ? s

Enter number of I/O addresses to read (1 to 4096): 1

Enter start address: 1

address Data


```
1:      0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
```

Example test of loop processor (LPM) data

This example reads an I/O parameter number 5 from control block number 3 in the LPM.

Enter LRN or device name of channel: chn50

Enter controller number(controller ID?) (required for Series 9000): 52

SERIES 9000 Control Block Input/Output

Enter function to be performed

- 0 - QUIT S90TST communications utility
- 1 - READ data
- 2 - WRITE data
- 3 - SELECT a new controller

Enter function: 1

Enter number of addresses to read (1 to 999): 1

Enter address[1] - block, index: 3,5

block	index	status list	data list
3	5	0x0000	60.00000

Troubleshooting communications errors

Error

0106 status I/O read error

Timeout for LRN xx , connection refused, where xx is the LRN number.

Remedy

- Check Series 9000 connection to the network.
- Check server connection to the network.

Error

010C no such file or directory

Indicating no Internet address configured.

Remedy

- Check the TCP/ IP hosts file for correct Internet address configuration.
- Check that the controller name matches the name in the TCP/ IP hosts database.
- Check the ELPM dip switch settings.
- Check the IP address definition for the controller in Quick Builder.
- Try using the TCP **ping** command to verify communication with the Series 9000, for example, `ping plcio` (using the TCP host name) or `ping 200.0.0.16` (using the IP address).

Viewing and modifying Series 9000 block parameters

You can view or modify Series 9000 block parameters after you have established communications with the server.

Make sure the channel and controller are enabled. On Station, use the System Status display for Channels and System Status display for Controllers. For information about calling up Station displays, see the *Configuration Guide* for your server.

To view a Series 9000 block parameter configuration:

- On the System Configuration display for Controllers, either double-click the name of the Series 9000 LPM, or select the LPM and go to the Station tool bar and click the Detail tool. The Controller Configuration display appears with the block configuration for the Series 9000. You can view any block with this display by typing **rec block_number**, where *block_number* is the number of the block you want displayed.
- On the Point Detail display on Station for a point parameter associated with the Series 9000 LPM parameter block, double-click the point name. Controller Slot Configuration displays appears with the block configuration for the Series 9000 LPM the point is associated with.

Use either display to modify editable parameter block values. You need to change your security level to ENGR or higher.

To modify a block parameter:

- Select the current value and enter a new value. The value is changed in both the server and the Series 9000 LPM.

Configuring Series 9000 set point programs and recipes



Note

This topic is only applicable if you have an Experion system.

If you are using Series 9000 controllers, you can use Station displays to view or edit set point programs or recipes.

Configuring Series 9000 set point programs (SPP)

You can have up to 99 set point programs in a single Series 9000 controller.

To view summary details of a controller’s set point program(s):

- 1 Select **Configure > S9000 > Set Point Programs**.
This calls up the Series 9000 Set Point Program Summary display.

Figure 4 Series 9000 set point program summary display

System Configuration		Set Point Programs		Series 9000 Controller		S9KCBIO	
		Set Point Programs		Recipes			
		Program Number	Description	Engineering Units	Number of Segments	Date	Modified Time
		1	SPP123	deg F	10	21-Nov-00	7:31:24
		2	DemoSPP	*C	10	06-Feb-01	9:29:46
		3	-		0		0:00:00
		4	-		0		0:00:00
		5	-		0		0:00:00
		6	-		0		0:00:00
		7	-		0		0:00:00
		8	-		0		0:00:00
		9	-		0		0:00:00
		10	-		0		0:00:00
		11	-		0		0:00:00
		12	-		0		0:00:00
		13	-		0		0:00:00
		14	-		0		0:00:00
		15	-		0		0:00:00
		16	-		0		0:00:00
		17	-		0		0:00:00
		18	-		0		0:00:00
		19	-		0		0:00:00
		20	-		0		0:00:00

- 2 From the **Series 9000 Controller** list, select the controller whose set point programs you want to view.
The lists contains all the controllers on Series 9000 channels that address LPM Control Block I/O data tables.

When you select a controller, all the Series 9000 set point programs are read from the selected controller, and the details are called up in the Series 9000 Set Point Program Summary display.

The display shows the following details for the selected controller:

- Program name(s)
- Units
- The number of segments that are being used by the program
- When the program was last modified



Note

The segments of a set point program are stored in blocks of 10, so the number of segments actually used is rounded up to the nearest 10. The Series 9000 can store a maximum of 100 10-segment blocks (that is, 1000 segments).

Viewing, editing, or creating set point programs

To view details of set point programs:

- Click the **Edit** button for a selected program on the set point program Summary display.

This calls up the Set Point Program Configuration display for that program.

Figure 5 Set point program configuration display

Segment	Type	Ramp (EU/min)	Setpoint (EU)	Soak Time (min)	Guaranteed Soak Band
1	RAMP	4.10	22.50	2:25:00	1.00
2	SOAK				
3	N/A	0.00	0.00	0:00:00	0.00
4	N/A				
5	N/A	0.00	0.00	0:00:00	0.00
6	N/A				
7	N/A	0.00	0.00	0:00:00	0.00
8	N/A				
9	N/A	0.00	0.00	0:00:00	0.00
10	N/A				

The Series 9000 Set Point Program Configuration display shows all the segments in a Series 9000 set point program. You can edit these segments or create new segments if you wish, but please note the following caution.



Caution

Before editing or creating set point programs, make sure that the Series 9000 software is not running. Editing set point programs while the Series 9000 software is running can cause errors in the Series 9000 software.

Before editing or creating set point programs, you must also ensure that the Series 9000 Configuration Software is not currently set as the Configuration Node. For details, see, “Releasing the Series 9000 as the configuration node” on page 36.

Releasing the Series 9000 as the configuration node

To release the Series 9000 configuration software as the Configuration Node:

- 1 Go to the Series 9000 software and select the controller from the list of **Configured Controllers**.
- 2 Click **Setup**.
- 3 Click **Release this PC as Configuration Node**.

Editing segments in a Series 9000 set point program

To edit segments (or create new segments) in a set point program, you must be logged on to Station at ENGR security level (or higher).

To edit the segments of the Series 9000, change the appropriate fields as necessary:

- Ramp Rate (in EU/min)
- Soak Time (hours : minutes : seconds)
- Set Point (EU)
- Guaranteed Soak Band (EU)
- Events

Creating new segments

To create a new pair of RAMP/SOAK segments, enter a ramp rate or soak time for the next segment.

You must create segments in the correct order: you cannot create a segment if the preceding segments have not yet been created. You can create a maximum of 100 segments (50 RAMP/SOAK pairs) for a Series 9000 set point program.

Deleting segments



Caution

Deleting a RAMP/SOAK segment will delete all subsequent segments in the program.

To delete a pair of RAMP/SOAK segments, set the ramp rate and soak time to zero.

Editing the program name or engineering units

To edit the Program Name or Engineering Units fields on the Set Point Program Configuration display, you need to:

- 1 Log on to Station at MNGR security level.
- 2 Ensure that the Series 9000 software has been released as the Configuration Node (as described in “Releasing the Series 9000 as the configuration node” on page 36).
- 3 Under the **Series 9000** heading in the top right of the Set Point Program Configuration display:
 - a. Set the server as the Configuration Node of the Series 9000. (Click the **Server is the Configuration Node** button to enable it.)
 - b. Set the **Mode** button to **Program**.



Note

If a redundant failover occurs, you need to repeat step a. once the new primary server is up and running, to ensure that the Configuration Node is set for the new primary server. (For more information about redundant systems and failovers, see the *Configuration Guide*.)

Restoring configuration settings

If you want to discard the changes that you have made in a set point program in the current editing session, you can use the **Restore** button on the Set Point Program Configuration display to restore the program configuration settings that were last downloaded.

Saving changes

If you want to save the changes that you have made in a set point program to the Series 9000, click the **Download** button on the Set Point Program Configuration display.

Returning to the summary display

From the Set Point Program Configuration display, you can return to the Set Point Program Summary display by clicking the  button beside the display name (in the upper right of the display).

Creating a new set point program

To create a new set point program:

- 1 Make sure you are logged on to Station as MNGR.
- 2 Click an unused line on the Series 9000 Set Point Program Summary display. This displays the Set Point Program Configuration display (as shown in “Set point program configuration display” on page 35).
- 3 In the **Program Name** field type a descriptive name for the new program.
- 4 Use the **Engineering Units** field to specify the engineering units to be used in the program.
- 5 Add segment data for the new set point program.
- 6 Click the **Download** button to write the new set point program to the controller.



Caution

If you leave a Set Point Program Configuration display without downloading the changes, any changes you have made will be lost.

Deleting a set point program

To delete a set point program, delete all data (segments, engineering units, and program name) from a set point program and then download it.

Resetting the Series 9000 Software as the Configuration Node

After editing (or creating new) set point programs and downloading the changes to the controller, you need to release the server as the Configuration Node and reset the Series 9000 configuration software as the Configuration Node.

To release the server as the Configuration Node:

- Disable the **Server is the Configuration Node** check box in the Series 9000 group box on the Set Point Program Configuration display (“Set point program configuration display” on page 35).

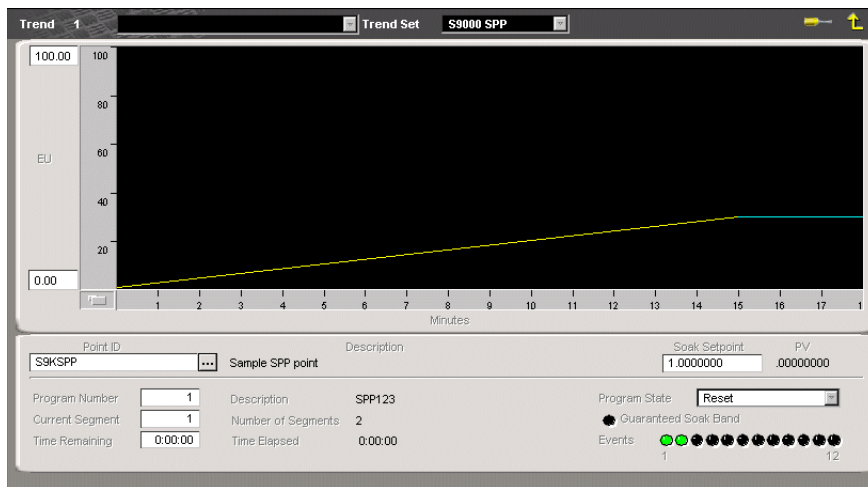
To reset the Series 9000 configuration software as the Configuration Node:

- 1 Go to the Series 9000 software.
- 2 Select the controller from the list of **Configured Controllers**.
- 3 Click **Setup**.
- 4 If the server is still the Configuration Node, select the **Force Release** option. Ensure there is no currently active configuration node before using the Force Release option.
- 5 Select **Set this PC to Configuration Node**.

Viewing Series 9000 set point program trends

Series 9000 set point programs can be viewed on a trend display, showing the set point program and highlighting the current segment when the program is running. The same set point program trend should not be called up simultaneously on two Stations.

Figure 6 Set point program trend display

**To use a set point program trend:**

- 1 Call up the Trend Summary display (click **Trend** on the System Menu display).
- 2 Click an unassigned Trend number to call up the Trend Configuration display.
- 3 Select **s9000 SPP** from **Trend Type**, and enter a point which has its PV scanning a Series 9000 SPP or SPP2 type block.

The SPP or SPP2 block contains a set point program number. The SPP or SPP2 block also contains parameters for the execution of the program. The state, current segment, soak set point, and soak duration can all be changed from this display if the security level of the Station is **SUPV** or higher and you have set the Series 9000 to **RUN** mode (see step 3 in “Editing the program name or engineering units” on page 37).

The following restrictions apply:

- The program number can only be changed when the state is **RESET**.
- The current segment and Soak Duration cannot be changed when the state is **RESET**.

**Note**

When these parameters are changed at execution time, the change is not permanent and the actual set point program is not changed, only the SPP block processing the current segment of the set point program.

Configuring Series 9000 recipes

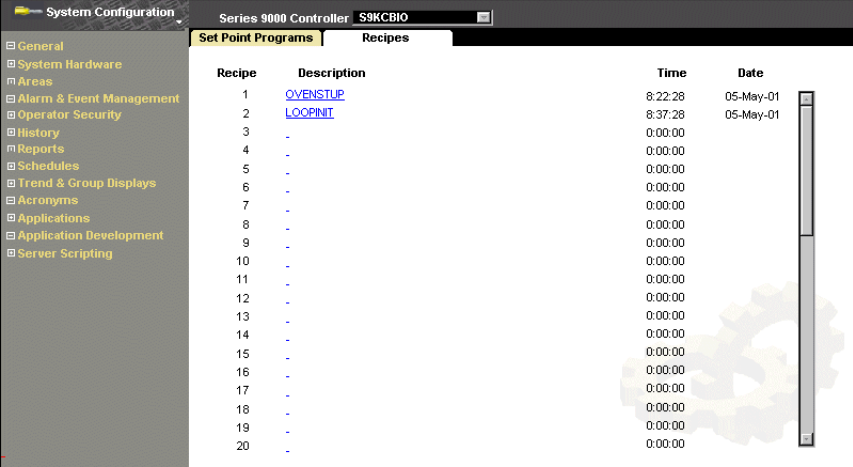
**Note**

Some servers, such as EBI, do not support recipes. See the *Configuration Guide* for details.

Recipes in a Series 9000 controller can be viewed and modified from the Series 9000 Recipe Summary display on Station.

To call up this display select **Configure > S9000 > Recipes**.

Figure 7 Series 9000 recipe summary



Recipe	Description	Time	Date
1	OVENSTUP	8:22:28	05-May-01
2	LOOPINT	8:37:28	05-May-01
3	-	0:00:00	
4	-	0:00:00	
5	-	0:00:00	
6	-	0:00:00	
7	-	0:00:00	
8	-	0:00:00	
9	-	0:00:00	
10	-	0:00:00	
11	-	0:00:00	
12	-	0:00:00	
13	-	0:00:00	
14	-	0:00:00	
15	-	0:00:00	
16	-	0:00:00	
17	-	0:00:00	
18	-	0:00:00	
19	-	0:00:00	
20	-	0:00:00	

The Series 9000 Recipe Summary shows a selected controller's recipe name and the time that the recipe was last modified. You can configure up to 50 recipes for a single Series 9000.

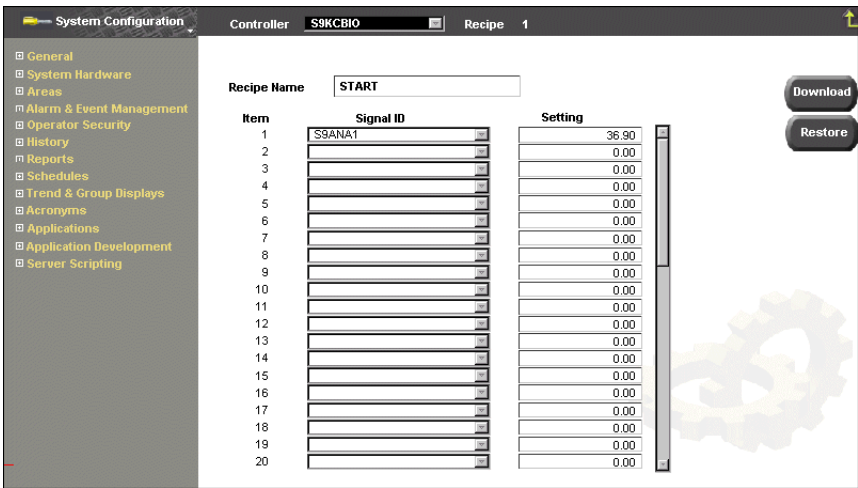
Use the **Controller** list to select the Series 9000 whose recipes you want to view or edit. The list contains all the controllers on Series 9000 channels which look at LPM Control Block I/O data tables. When you select a controller, all the Series 9000 Recipes are read from that controller. This may take a few seconds.

To configure a recipe:

- 1 Use the **Date** and **Time** fields to specify when the process is to run.
- 2 Use the Recipe Configuration display to view, edit, or create Series 9000 recipes ("Series 9000 recipe configuration display" on page 42).

To call up this display, click the **recipe number or description** on the Recipe Summary display.

Figure 8 Series 9000 recipe configuration display



The Series 9000 Recipe Configuration display shows details of a Series 9000 Recipe.

You can edit these details as necessary, if the security level of your Station is **supv** or higher.

Editing recipe items

To edit an existing Series 9000 Recipe item.

- 1 Go to the Series 9000 Recipe Configuration display (“Series 9000 recipe configuration display” on page 42) for the recipe you want to change.
- 2 Do one or both of the following:
 - Change the current selection in the **Point ID** combobox.
 - Change the value in the **Setting** field for that item.
- 3 To write the changes that you have made in the recipe to the Series 9000, press the **Download** button on the Series 9000 Recipe Configuration display.



Caution

If you leave a Series 9000 Recipe Configuration display without first downloading the changes, any changes you have made will be lost.

Creating new recipe items

You configure up to 50 items in a Series 9000 recipe.

To create a new recipe item:

- 1 Go to an unused **Item** field on the Series 9000 Recipe Configuration display (“Series 9000 recipe configuration display” on page 42) and make a selection from the **Point ID** combobox.
- 2 Enter a value in the **Setting** field.
You cannot assign a value unless you made a selection for the **Point ID**.
- 3 To write the changes that you have made in the recipe to the Series 9000, press the **Download** button on the Series 9000 Recipe Configuration display.

Deleting recipe items

To delete a recipe item, set the **Point ID** combobox to be blank (that is, the first element in the combobox list). This also sets the corresponding **Setting** value to zero.

Restoring recipe settings

If you change your mind about changes that you have made in a Recipe in the current editing session, you can use the **Restore** button on the Series 9000 Recipe Configuration display to restore the recipe configuration settings that were last downloaded.

Returning to the summary display

To go back to the Series 9000 Recipe Summary display, use the  button on the right of the title bar.

Creating a new recipe

To create a new recipe:

- 1 Go to an unused Recipe entry on the Series 9000 Recipe Summary display, and click the number or description area of an unassigned recipe number.
This calls up the Series 9000 Recipe Configuration display (as shown in “Series 9000 recipe configuration display” on page 42).
- 2 Use the **Recipe Name** field to enter a descriptive name for the new recipe.
- 3 Add items as described in “Creating new recipe items” on page 43.
- 4 Click the **Download** button to write the recipe to the controller.

Deleting a recipe

To delete a recipe, clear all **Point ID** selections on the Series 9000 Recipe Configuration display for that recipe, and then download it.

Index

A

addressing
 I/O addressing example 23
 LPM I/O and configuration data 22
 register address examples 20
 S9000 PLC registers and digitals 19
 Series 9000 controller 19
architecture 11

C

channel definition, for Series 9000 14
communication
 errors, troubleshooting for S9000 32
 testing 30
composite mode, support in PID blocks 24
configuration method, choosing 12
Configuration Node, setting 36
Configuration Software, using with Quick
 Builder 18
configuring S9000, steps for 5
controller
 addressing S9000 19
 definition for Series 9000 16

D

data formats, scaling with 24
documentation, for Series 9000 6

E

ELPM IP address, setting 10
errors, S9000 communication 32
Events, Series 9000 36

G

Guaranteed Soak Band, Series 9000 36

L

LAN port definition 15
LPM I/O and configuration data, addressing 22

O

operator tag, support 23

P

PID blocks, support for composite mode 24
PLC registers and digitals, addressing 19
Port tab 15
port type, S9000 LAN 15

Q

Quick Builder, using with the S9000
 Configuration Software 18

R

Ramp rate, Series 9000 36

S

scan packets 26
scanning
 optimizing for Series 9000 26
Series 9000
 configuration methods 12
 Configuration Node setting 36, 37, 38
 configuring Recipes 40
 configuring Set Point Programs 34
 other documentation 6
 scan packets 26
Set Point Programs
 Series 9000 configuration 34
Soak time, Series 9000 36

T

testing

example test of S9000 LPM data 31

example test of S9000 PLC data 30

Honeywell Safety Manager Integration Reference

Issue

1

Date

November 2005

Notice

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 Limited Australia.

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.

Copyright 2005 – Honeywell Limited Australia

Honeywell trademarks

PlantScape[®], SafeBrowse[®], **TotalPlant**[®] and TDC 3000[®] are U.S. registered trademarks of Honeywell International Inc.

Experion[™] and Honeywell Enterprise Buildings Integrator[™] are trademarks 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.

Support and other contacts

For technical assistance, call your nearest Honeywell office.

Contents

1	Getting started	5
	Support and documentation for Safety Manager	6
	Documentation for Safety Manager	6
	Safety Manager-specific terms	7
2	Safety Manager controller setup	9
	Differences between serial and Ethernet controllers	10
	Supported architectures	11
	Communication settings	14
	Ethernet settings	14
	Assigning SOE numbers for Sequence of Events	16
	Reserved SOE numbers	16
	Interpreting SOE lines	17
3	Safety Manager controller configuration and addressing	19
	Defining a Safety Manager channel	20
	Safety Manager channel main properties	20
	Safety Manager port properties	22
	Defining a Safety Manager controller	25
	Safety Manager controller main properties	25
	Defining a Safety Manager address for a point parameter value	28
	Entering an address	28
	Data formats support for Ethernet controllers	30
	Optimizing scanning performance	33
4	Server and Station Tasks for Safety Manager	35
	Accessing diagnostics for an Ethernet controller	36
	System information	36
	Extended diagnostics	36
	Testing communications with the server	37
	Testing communications	37
	Troubleshooting Safety Manager communication errors for an Ethernet controller. . .	37

Getting started

1

This reference describes how to set up, configure, and test Safety Manager communications with the server. There is detailed information for defining the Safety Manager using Quick Builder. You are expected to have some experience with server and Safety Manager configuration.

Complete each step before commencing to the next.

Steps for connecting and configuring Safety Manager.

Steps	Go to:
Determine Safety Manager configuration	page 11
Setting up the communications parameters using the Safety Manager configuration software	page 14
Use Safety Manager configuration software to set SOE numbers	page 16
Define channels with Quick Builder	page 20
Define controllers with Quick Builder	page 25
Download channel and controller definitions to the server	
Test communications	page 37
Define points with Quick Builder	page 28

Support and documentation for Safety Manager



Note

Early releases of Safety Manager do not support serial communications or Sequence of Events reporting. All references to serial communications and Sequence of Events reporting in this document should be ignored for these controllers.

The server supports Safety Manager controllers based on serial communication links and Ethernet based Safety Manager controllers.

The server communicates with:

- Ethernet controllers using a proprietary protocol based on the Modbus protocol.

For more information on controller types, see “Differences between serial and Ethernet controllers” on page 10.

Documentation for Safety Manager

This reference only provides supplemental information to interface Safety Manager controllers with the server. For detailed information about installing and configuring Safety Manager controllers, see the Safety Manager documentation.

Safety Manager-specific terms

Base SOE number

The unique ID assigned to a serial (non-Ethernet) controller so that the server can extract SOE data from that controller.

Control processor

Core component of the Safety Manager controller consisting of a power supply unit (PSU), quadruple processor pack (QPP) and 1 or 2 communication modules (SMM or USI).

Safety Manager controller

The controller of Safety Manager. The default Safety Manager controller has a redundant control processor.

SOE

Sequence of Events.

Safety Manager controller setup

2

This chapter describes how to set up a Safety Manager controller for the system.
The tasks for setting up a Safety Manager controller are:

For	Go to:
Differences between serial and Ethernet controllers	page 10
Supported architectures	page 11
Communication settings	page 14
Using Safety Manager configuration software to assign SOE numbers	page 16

Differences between serial and Ethernet controllers

The server supports:

- **Serial controllers:**
Safety Manager controllers based on serial communication and
- **Ethernet controllers:**
Safety Manager controllers based on Ethernet communication.

These two controller types differ in the following ways:

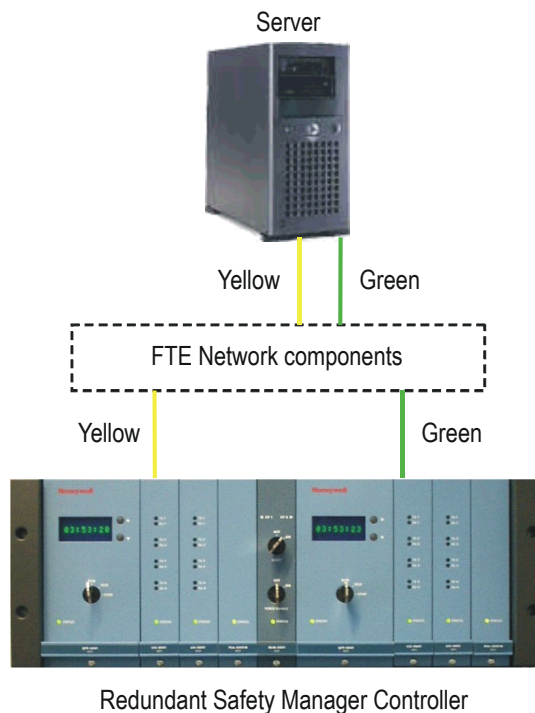
- Serial controllers use Modbus; whereas Ethernet controllers use a proprietary protocol based on Modbus
- The diagnostic scan rate is not configurable in Ethernet controllers
- The Base SOE is not required for Ethernet controllers
- There is a separate test utility for each type of controller
- System Information and Extended Diagnostics are available for Ethernet controllers

Supported architectures

The server communicates with the Ethernet channels of the Safety Manager controller's USI-0001 modules using a proprietary protocol based on the Modbus protocol.

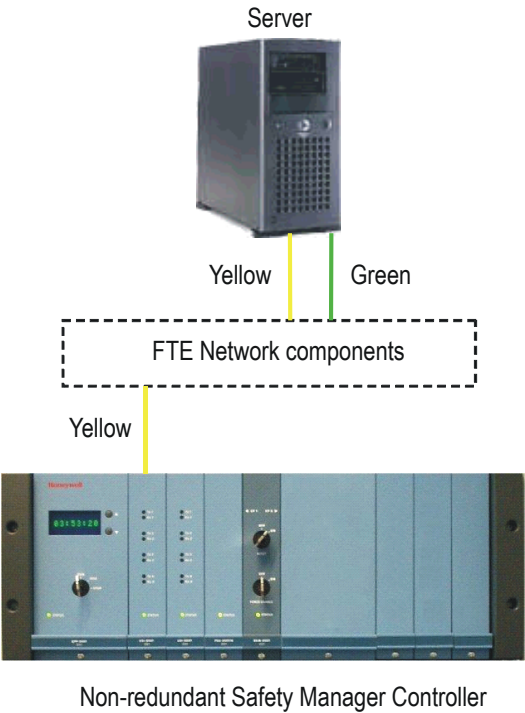
To connect a redundant Safety Manager controller to an FTE network, CP1 is connected to the yellow switch and CP2 is connected to the green switch, as shown in the following figure.

Figure 1 Redundant Safety Manager in an FTE network



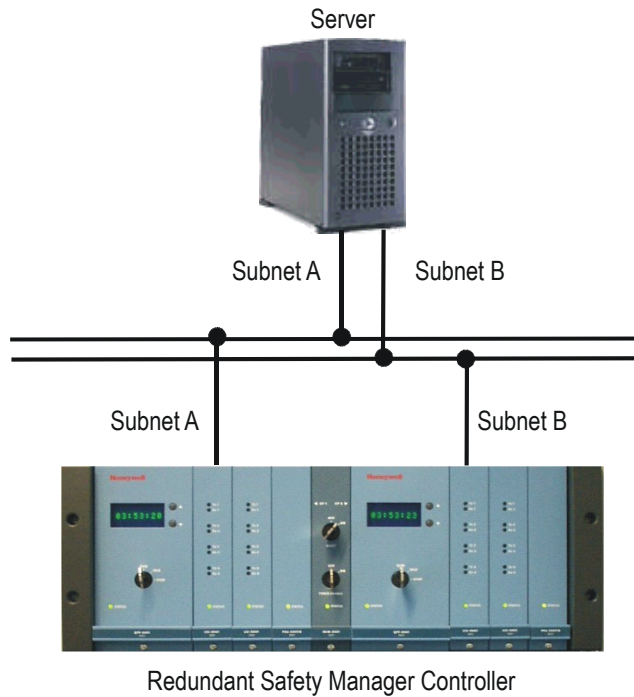
To connect a non-redundant Safety Manager controller to an FTE network, CP1 is connected to the yellow switch, as shown in the following figure.

Figure 2 Non-redundant Safety Manager in an FTE network



To connect a redundant Safety Manager controller to a non-FTE redundant network, CP1 is connected to the subnet A and CP2 is connected to subnet B. Safety Builder R101.2 or later is required in order to set the IP address of CP1 independently of CP2.

Figure 3 Redundant Safety Manager in a non-FTE redundant network



Attention

Connecting or configuring a single Ethernet Safety Manager controller to multiple servers is not a supported configuration.

Communication settings

The communication settings are specific to the controller type.

Ethernet settings

The Ethernet connection is made to an available Ethernet channel of the controller’s USI module.

Communication address

To communicate with an Ethernet Safety Manager controller, the server uses a Communication address, which can be calculated using the Safety Manager System number specified when configuring the controller.

You use the Safety Manager configuration utility, Safety Builder, to specify the Safety Manager System number, which must be unique within a project of Safety Managers.



Note

The Communication address equals 4 times the Safety Manager System number.

Ethernet data tables

Ethernet Safety Manager controllers use the following Modbus data tables.

In Quick Builder, you need to define a separate “logical” controller for each data table to which the server needs access. For example, if the server needs access to the Coils and Registers, you need to define two logical controllers.



Note

The following points can only be built on a SOE only controller:

- Safety Manager variables with a SOE sequence number, but without a DCS address.
- Safety Manager system events. See “Reserved SOE numbers” on page 16.

Table 1 Ethernet data tables

Data Table	Server Addressable Range (DCS address)
Coils (Safety Manager types: BI, BO)	1 - 8192
SOE Only	N/A

Table 1 Ethernet data tables

Data Table	Server Addressable Range (DCS address)
Registers (Safety Manager types: AI, AO)	10001 - 18192

Assigning SOE numbers for Sequence of Events

Safety Manager controllers have SOE (Sequence Of Events) capabilities. If you want the server to record a controller's SOEs in its SOE log, you need to assign an SOE number to each internal address.



Note

In the case of a serial Safety Manager controller, you need to define a Base SOE for the controller.

To record a controller's SOEs in its SOE log:

- 1 Assign an SOE number to each internal address you want the server to record. (It is recommended that you assign SOE numbers in a tight block for each controller.)
You use the Safety Manager configuration utility, Safety Builder, to assign SOE numbers.
- 2 When defining the controller in Quick Builder, select the **SOE Enable** check box. (If you define several "logical" controllers, only select this for one controller.) See "Defining a Safety Manager controller" on page 25.
- 3 Assign a separate point, in Quick Builder, to each SOE number. A point can only have 1 SOE number associated with it.

Reserved SOE numbers

Safety Manager controllers reserve the following SOE numbers for system events. If you want the server to log these system events, you need to assign points to them that are reserved for system events.

SOE Number (Ethernet)	System Event
0	All forces cleared
1	Safety Manager System Fault
2	SOE buffer full
3	SOE buffer Empty
5	Force Event


Note

- 4 is not a valid event.
- The “SOE Buffer Empty” event cannot be stored in the server SOE log. If a point is defined with this SOE number, it is ignored.
- To be logged on to the SOE display when using an Ethernet Safety Manager controller, the System Events must be built on a SOE only controller. This is because System Events have no DCS address associated with them.

Interpreting SOE lines

The following fields of the server’s SOE file are populated for each SOE with a matching SOE number:

Field	Description
Date & Time	The date and time given to the SOE by the controller.
Source	The server’s point ID that corresponds with this SOE.
Condition	The Safety Manager SOE number associated with this SOE.
Action	Only applicable to Ethernet controllers. Information indicator dependent on the types of SOE. See “Action descriptions” on page 17.
Description	The server’s point description.
Value	The value returned in the SOE (if applicable). For a status point built against a coil, this displays the appropriate state descriptor for that point. Indeterminate values display either blank or zero.

Action descriptions

Table 2 Action descriptions

Event Type	Safety Manager Data Type	Safety Manager Description	Server-action Field
Event Report*	Any		N/A**
Process Variable Event	Boolean	True to False	F
		False to True	T
	Analog	Low	LO
		Healthy	HLT

Table 2 Action descriptions (*continued*)

Event Type	Safety Manager Data Type	Safety Manager Description	Server-action Field
		High	HI
Process Variable Force Event	Boolean	Force Set	SET
		Force Cleared	CLR
	Analog	Force Set	SET
		Force Cleared	CLR
Diagnostic Event	Diagnostic Event		N/A

* System events. See “Reserved SOE numbers” on page 16.

** Actions that are not applicable or indeterminate are displayed as “--”.

Safety Manager controller configuration and addressing

3

This chapter describes how to configure a Safety Manager controller for your system using Quick Builder.

The Quick Builder controller configuration tasks are described:

For:	Go to:
Defining a Safety Manager channel	page 20
Defining a Safety Manager controller	page 25
Defining an address for a point parameter value	page 28
Creating points in Quick Builder after downloading channels and controllers	page 28
Optimizing scanning performance	page 33

Defining a Safety Manager channel



Note

If you use both serial and Ethernet Safety Manager controllers, you need to create a separate channel for each type of controller.

To define a channel using Quick Builder:

- 1 Click  to add a channel.
The **Add Items** dialog box opens.
- 2 Select **Channel** as the item and **Safety Manager** as the channel type.
- 3 Enter the channel details on the **Main** tab for the channel. See “Safety Manager channel main properties” on page 20.
- 4 Click the **Port** tab, and select a port type appropriate to the controller type.
Define the other properties as required. See “Safety Manager port properties” on page 22.

Safety Manager channel main properties

Use the Main tab to enter the basic channel properties for a Safety Manager channel.

Property	Description
Name	A unique name for the channel. A maximum of 10 alphanumeric characters. (Double quotation marks and spaces are not allowed.)
Description	(Optional) A description of the channel. A maximum of 30 characters can be used, including spaces.

Property	Description
Marginal Alarm Limit	The communications alarm marginal limit at which the channel is declared to be marginal. When this limit is reached, a high priority alarm is generated. A channel barometer monitors the total number of requests and the number of times the controller did not respond or response was incorrect. The barometer increments by 2 or more, depending on the error and decrements for each good call. To calculate an acceptable limit, multiply the square root of the number of controllers on the channel by the Marginal Alarm Limit defined for those controllers. (Normally, you specify the same value for all controllers on a channel). For example, if there are 9 controllers on the channel and their Marginal Alarm Limit is set to 10, the value would be $[\sqrt{9} \times 10 = 30]$.
Fail Alarm Limit	The communications alarm fail limit at which the channel is declared to have failed. When this barometer limit is reached, an urgent alarm is generated. Set this to double the value specified for the channel Marginal Alarm Limit.
Connect Timeout	The time (in seconds) the server waits to connect to the controller before abandoning the connection. The default is 10 seconds. Use the default value unless the communications line has a high error rate or unless you are using modems.
Read Timeout	The time (in seconds) that the server waits for a reply from the controller. The default is 2 seconds. Use the default value unless the communications line has a high error rate or unless you are using modems.
Item Type	Shows the type of item specified when this item was created.
Last Modified	Shows the date of the most recent modification to this channel's property details.
Item Number	The unique item number currently assigned to this item by Quick Builder. You can change the item number if you need to match your current server database configuration. The item number must be between 1 and the maximum number of channels allowed for your system.

Safety Manager port properties

The Port tab defines the communication-related properties for a channel. The **Port Type** for Safety Manager controllers can be:

- **serial**. Only applicable to a directly-connected serial Safety Manager controller. A serial communications interface, such as RS-232. See “Serial port properties” on page 22.
- **TerminalServer**. Only applicable to a serial Safety Manager controller that communicates with the server via a terminal server. See “TerminalServer port properties” on page 23.
- **LANVendor**. Only applicable to an Ethernet Safety Manager controller. See “LANVendor port properties” on page 24.

A communications link being used as a redundant link requires an additional port definition for the redundant port. After you complete the port definition, enter the same kind of port definition for the redundant port.

Serial port properties



Note

Only applicable to serial Safety Manager controllers.

Property	Description
Serial Port Name	The device name of the serial port.
Baud Rate	The number of data bits per second. The default is 9600.
Number of Data Bits	The number of data bits used for transmission. The default is 8.
Stop Bits	The number of stop bits used for transmission. The default is 1.
Parity	The parity verification used on the port. The default is NONE.
Checksum	The type of checksum error detection used for the port. Select the value that matches the setting on the communication device: • CRC16_0 or CRC16_1 (if Cyclic Redundancy Check (CRC) is set) • ONESCOMP or TWOSCOMP (if Longitudinal Redundancy Check (LRC) is set) • XOR (If exclusive or is set)

Property	Description
XON/XOFF	The type of XON/XOFF software flow control used to stop a receiver from being overrun with messages from a sender. The types are: • None (default) • Input (use XON/XOFF to control the flow of data on the receive line) • Output (use XON/XOFF to control the flow of data on the transmit line)
Handshaking Options	For RS-232 • Enable RTS/CTS flow control. Stops a receiver from being overrun with messages from a sender by using RTS/CTS flow control. • Detect DCD. Select if the Data Carrier Detect communication status line of the COM port requires monitoring (usually when using modem or microwave linking). When selected, the communications fails if the desired COM status line is not high—for example, on a dial-up link connection for a modem. • Detect DSR. Select if the Data Set Ready communication status line of the COM port requires monitoring (usually when using modem or microwave linking). When selected, the communications fails if the desired COM status is not achieved. Note: No options available for RS-422. For RS-485. The server does not support RS-485 for Safety Manager controllers.

TerminalServer port properties



Note

Only applicable to serial Safety Manager controllers.

Property	Description
Terminal Server TCP Host Name	The name and port number of terminal server to which the channel is connected.

Property	Description
Terminal Server TCP Port No	You can specify either a TCP host name or an IP address, but it must match the TCP host name used when you installed and internally configured the terminal server.
Idle Timeout	The time, in seconds, the channel waits for a successful connection to the server before closing the connection. A value of 0 indicates that the connection is never closed.
Checksum	The type of checksum error detection used for the port. Select the value that matches the setting on the communication device: • CRC16_0 or CRC16_1 (if Cyclic Redundancy Check (CRC) is set) • ONESCOMP or TWOSCOMP (if Longitudinal Redundancy Check (LRC) is set) • XOR (If exclusive or is set)

LANVendor port properties



Note

Only applicable to Ethernet Safety Manager controllers.

Property	Description
Port Name	Leave blank.

Defining a Safety Manager controller

You need to define a controller to describe each logical controller on a channel.

To define a controller using Quick Builder:

- 1 Click  to add a controller to display the Add Items dialog box
- 2 Select `Controller` as the item and `Safety Manager` as the controller type.
- 3 Select the appropriate channel from the **Channel Name** list. (When you select the channel, the appropriate serial or Ethernet properties appear.)
- 4 Define the controller's other properties. See "Safety Manager controller main properties" on page 25.

Safety Manager controller main properties

Use the Main tab to define the basic properties for a Safety Manager controller.

Property	Description
Name	The unique name for the controller. A maximum of 10 alphanumeric characters. (Double quotation marks and spaces are not allowed.)
Description	A description for this controller. A maximum of 30 characters, including spaces.
Channel Name	The channel on which the controller communicates with the server. If you use both serial and Ethernet controllers, you must select a serial/terminal server channel for a serial controller, and an LAN Vendor channel for an Ethernet controller.
Marginal Alarm Limit	The communications alarm marginal limit at which the controller is declared to be marginal. When this value is reached, a high priority alarm is generated. This limit applies to the controller barometer which monitors the total number of requests to the controller and the number of times the controller did not respond or response was incorrect. The barometer increments by 2 or more, depending on the error and decrements for each good call. Type a new value in this field if you do not want the default, 25.

Property	Description
Fail Alarm Limit	The communications alarm fail limit at which the controller is declared to have failed. When this value is reached, an urgent alarm is generated. Set this to double the value specified for the controller Marginal Alarm Limit.
Communication Address	Set this to 4 times the Safety Manager System number. The Safety Manager System number is set using the Safety Manager configuration utility, Safety Builder.
Safety Manager Type	Shows the controller type, which corresponds to the type of channel (serial or Ethernet) selected in Channel Name.
Data Table	The data table that this controller addresses. See: “Ethernet data tables” on page 14 for an Ethernet controller
Diagnostic	Only applicable to a serial controller. The value, in seconds, between diagnostic polls. The default is 60 seconds. To disable diagnostic polls, set the value to 0.
Base SOE	Only applicable to a serial controller. The base address for this controller. See “Assigning SOE numbers for Sequence of Events” on page 16.
IP Address 1 IP Address 2	The controller’s Ethernet addresses.
SOE Enable SOE Rate	Select SOE Enable if you want SOE data. Specify the time, in seconds, between polls for SOE data in SOE Rate. Only one logical controller per physical Safety Manager controller can have SOE Enabled. If an SOE only controller is built, this should be the only logical controller with SOE enabled for that particular physical Safety Manager controller. The SOE Rate must be a minimum of 5 seconds and defaults to 30 seconds. See “Assigning SOE numbers for Sequence of Events” on page 16.
Synch Enable Sync Time	Select Synch Enable if you want to synchronize the controller time with the server time. Specify the time, in minutes, since midnight before synchronizing in Synch Time. The default is –1 (no synchronization). See “Time synchronization” on page 27.

Property	Description
Item Type	Shows the type of item specified when this item was created.
Last Modified	Shows the date of the most recent modification to this channel's property details.
Item Number	The unique item number currently assigned to this item by Quick Builder. You can change the item number if you need to match your current server database configuration. The number must be between 1 and the maximum number of controllers allowed for your system.

Time synchronization

You can synchronize the time of Safety Manager controllers with the server provided they are configured to accept the time from the server.

The **Synch Time** parameter is specified in minutes since midnight. When the assigned time is reached, the server sends a synchronization command with the current time to the controller.

If you define several “logical” controllers for a particular controller, you can specify a different synchronization time for each logical controller, in order to synchronize the physical controller at different intervals in the day.

Defining a Safety Manager address for a point parameter value

Entering an address

The format for a Safety Manager data table address is:

ControllerName Location

Part	Description
<i>ControllerName</i>	The name of the Safety Manager controller.
<i>Location</i>	The location in the controller where the value is recorded. See “Location syntax for coils and registers” on page 28 for details.

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

Location syntax for coils and registers

Address [Format|BitNumber] [SOE SOENumber]

Part	Description
<i>Address</i>	The server addressable memory address within the controller’s data table. For details, see: • “Ethernet data tables” on page 14 (Ethernet controller)

Part	Description
<i>Format</i>	(Optional) Select an appropriate format to read raw values. To scale See “Scaling with data formats” on page 31. To read without scaling Select a format of C16, or enter a 0, to read all 16 bits in the register without scaling. To read less than 16 bits without scaling enter the starting bit number (1 to 15). If you are not using scaling, the point range is still used for PV indicator bar height only (the PV indicator bar is on the Point Detail display on Station). To read floating point values The supported floating point formats are specific to the controller type: - Serial, see “IEEEFP formats for serial controllers” on page 30. - Ethernet, see “Data formats support for Ethernet controllers” on page 30. Select format IEEEFP to read two consecutive registers as a single precision floating point number.
<i>BitNumber</i>	(Optional) For Input register and Holding register data tables, a starting bit number can be specified. The valid range is 0 (default) to 15 where 0 is the right most bit in the 16-bit register. Analog and accumulator point parameters can read up to 16 bits. Status point parameters read 1, 2, or 3 consecutive bits.
<i>SOENumber</i>	The SOE number. See “Assigning SOE numbers for Sequence of Events” on page 16.



Note

For a status point with an OP configured with four output states (that is, across two consecutive coils), control of the OP is carried out by executing separate writes to each of the two coils. Therefore, a value is written to the first coil then another immediately to the second coil. Safety Manager logic should be programmed to make allowance for these two separate writes operations.

Example**Analog point**

PV source 161 U4095 SOE 1000
 SP destination 162 U4095
 Mode destination 25 1

Status point

PV source 26

Accumulator point

PV source 171 C16

IEEEFP formats for serial controllers

Format	Description
IEEEFPB	Bytes are big endian format (this is the same as IEEEFP)
IEEEFPBB	Bytes are byte-swapped big endian format
IEEEFPPL	Bytes are little endian format
IEEEFPPLB	Bytes are byte-swapped little endian format

**Note**

- IEEEFP numbers use two data addresses to hold the number, address and address -1. Do not specify address -1 (the lower address) as the point parameter location address.
- When configuring data tables, do not assign overlapping data addresses if floating point values are used and do not set an IEEEFP address to 1.

Data formats support for Ethernet controllers**Safety Manager types: BI, BO**

Format	Description
S8B	Short signed integer
S16B	Signed word
S32B	Signed long integer

Format	Description
IEEEFP	IEEE Floating point

Safety Manager types: AI, AO

Format	Description
SM020MA	0 - 20 mA
SM420MA	4 - 20 mA
SM05V	0 - 5 V
SM15V	1 - 5 V
SM010V	0 - 10 V
SM210V	2 - 10 V

Scaling with data formats

Parameter values with addresses in the Input register and Holding register data tables can be scaled with a data format. Select the format that corresponds to the counts that have been set in the controller register.



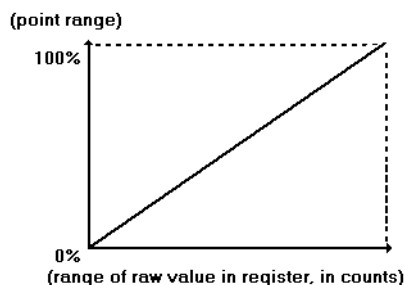
Note

If auxiliary parameters have a data format type that requires scaling (U4095, U999, and so on), they will take the same range as the PV.

The data format tells the server how to interpret the register value. Raw values in the controller register tables can be scaled by the 0% and 100% point range values in order to convert them into engineering units (EU).

To select a format for scaling, you select the format that corresponds to the counts that have been set in the register where the point parameter value is sourced.

Figure 4 Scaling Raw Data



Location syntax for SOE only points

POS: *Index* SOE *SOENumber*

Part	Description
<i>Index</i>	The server's internal position allocated to this point. It must be unique. Range = 1 - 8192
<i>SOENumber</i>	The SOE number. See "Assigning SOE numbers for Sequence of Events" on page 16.

Optimizing scanning performance

The maximum amount of data that can be acquired from a Safety Manager controller is influenced by the rate of sending scan packets to the controller. An understanding of Safety Manager scan packets will help you configure points so that optimal data acquisition performance is achieved, by maximizing the amount of data acquired with each scan packet.

The scan packets that have been built can be listed by using the list scan utility, **lisscn**. Listing scan packets helps verify the scanning strategy. See the *Configuration Guide* for usage of **lisscn**.

Server and Station Tasks for Safety Manager

4

This chapter describes tasks for the Safety Manager controller that you perform either on the server or from any Station.

For:	Go to:
Accessing diagnostics for an Ethernet controller	page 36
Testing communications with the server	page 37

Accessing diagnostics for an Ethernet controller

System information and extended diagnostic data are accessible from Station for Ethernet Safety Manager controllers.

System information

To access Safety Manager System Information, double-click the PV of an Ethernet controller point. Alternatively, click **Details** in the controller Status, which shows the **Ext Diagnostics** tab, then click the **System Info** tab for the required link.

The system information and extended diagnostic information is updated automatically every 50 seconds. You can manually refresh the information by clicking on the **Refresh** button.

If the diagnostic data cannot be retrieved from the Ethernet controller, the most recent diagnostic data from the controller will continue to be displayed until the new data is obtained. If successive attempts to obtain diagnostic data from the controller fail, then an alarm message will be generated in the Alarm Summary.

Extended diagnostics

To access Safety Manager Extended Diagnostics, click the **Details** button in the controller Status. This shows up to 32 extended diagnostics messages for each Ethernet controller. Each message represents a hardware or software fault associated with that particular controller.

If an extended diagnostic message is present for a controller, an alarm with a value of `message(s) Available` will be present in the Alarm Summary. To see the Extended Diagnostic information for an alarm from the Alarm Summary, double-click the alarm to call up the controller Status, then click the **Details** button.

Testing communications with the server

Testing communications

A diagnostic utility, **smpksetst** (for Ethernet controllers), is included as part of the server software. This utility tests communications between the server and a Safety Manager controller. Channels and controllers must be downloaded from Quick Builder before testing.

Before using the utility, make sure that:

- You have set up your Safety Manager controllers according to their user manuals.
- All cables are connected.
- You have configured your channels in Quick Builder without error and downloaded all configuration information to the server without error.
- The server need not be running while using the utility as long as the database service is running. If making a connection through a terminal server, the server daemon service should also be running.
- The server is not communicating with your controllers. The test utilities might interfere with communications.

To stop the server, enter the command at the command prompt:

hscserver /load

Answer 'y' to every prompt. This unloads the server, but leaves the database in memory.

Using smpksetst

To use **smpksetst**, start a command prompt window and type: **smpksetst**.

The test utility is menu-driven, and self-explanatory. If there are any communications errors, appropriate error messages will describe what is wrong.

Troubleshooting Safety Manager communication errors for an Ethernet controller

Ensure scan packets are optimized. For more information, see "Optimizing scanning performance" on page 33.

If there are several "logical" controllers on a channel, make sure that there is no more than one which has SOE enabled.

Index

A

addressing
 location syntax 28, 32
 using Quick Builder 28

B

Base SOE 16

C

channel, defining in Quick Builder 20
communication address
 Ethernet 14
communication settings 14
communications, testing 37
configuration utility 14
configuring Safety Manager, steps for 5
controller
 addressing data tables 28
 defining in Quick Builder 25

D

data formats, scaling 31
diagnostic utilities, using 37
documentation for Safety Manager 6

E

Ethernet
 communication address 14
 Safety Manager system number 14
Extended Diagnostics 36

L

LANVendor port, defining for Safety
Manager 24

P

port definition
 LANVendor 24
 serial 22
 terminal server 23
Port tab 22

S

Safety Manager system number
 Ethernet 14
scanning, optimizing 33
Sequence of Events 16
serial port, defining for Safety Manager 22
server tasks for Safety Manager 35
smpksetst test utility, using 37
SOE
 described 16
 lines 17
SOE number
 described 16
 reserved 16
System Information 36

T

terminal server port, defining for Safety
Manager 23
time synchronization 27

Honeywell TDC 3000 Data Hiway Interface Reference

Issue

5.1

Date

November 2005

Notice

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 Limited Australia.

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.

Copyright 2005 – Honeywell Limited Australia

Honeywell trademarks

PlantScape[®], SafeBrowse[®], **TotalPlant**[®] and TDC 3000[®] are U.S. registered trademarks of Honeywell International Inc.

Experion[™] and Honeywell Enterprise Buildings Integrator[™] are trademarks 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.

Support and other contacts

For technical assistance, call your nearest Honeywell office.

Contents

1	Getting started	1
	Referenced documentation	2
	Glossary of terms.	3
2	System comparison	1
	Overview	2
	TDC box support.	5
	Operational considerations	9
3	Data Hiway interfaces	1
	Data Hiway Ethernet Bridge	2
	Data Hiway Ethernet Bridge architectures	2
	Fault Tolerant Ethernet architecture	3
	DHEB hardware	4
	DHEB hardware setup.	4
	DHEB connections	5
	DHEB grounding considerations.	5
	DHEB software installation.	6
	DHEB software configuration.	6
	DHEB troubleshooting	9
	Data Hiway Bridge	10
	Data Hiway Bridge architectures.	10
	DHB hardware.	12
	DHB hardware setup	12
	Hiway Bridge switch settings	13
	DHB connections.	15
	DHB grounding considerations.	18
	DHB troubleshooting.	19
4	System configuration	1
	Defining TDC channels.	2
	Channel Main tab settings	2
	TDC channel port properties	4
	Channel LANVendor port settings	4
	Channel serial port settings	4
	Channel terminal server port settings	5
	TDC channel redundant port properties	5
	Channel configuration	6

Defining TDC controllers	7
Controller Main tab settings	7
Testing the communications	9
TDC I/O slot definitions	9
TDC points	11
Introduction	11
Exception scanning	12
Optimizing the scanning	15
Defining TDC points	15
TDC point property settings	16
TDC parameter addresses	21
Basic Controller (CB) addressing	22
Extended Controller (EC) addressing	23
Multifunction Controller addressing	25
Process Interface Unit (PIU) addressing	30
Data Hiway Port (DHP) and LCS 620 HIM addressing	32
Hardware generic addressing	33
TDC engineering unit conversions	35
Special TDC point and group displays	36
Additional tasks	39
5 System operations	1
Monitoring the system status	2
TDC Data Hiway commands	4
Data Hiway switch command	4
Basic Controller (CB) and Extended Controller (EC) commands	4
Multifunction Controller (MC) commands	5
Reserve Controller Director (RCD) commands	5
PIU, DHP and HIM commands	5
TDC error codes	7
Hiway addressing/communication error codes	8
HLPIU (3xx), LLPIU (4xx) and LEPIU (5xx) error codes	13
DHP or 620 HIM error codes	17
Basic controller primary (1xx) and reserve (6xx) error codes	21
Extended controller primary (7xx) and reserve (8xx) error codes	23
Multifunction controller (MFC and AMC) error codes	27
UAC primary controller codes	36
UAC reserve controller Director CB (6xx) and EC (8xx) error codes	38
UAC reserve controller Director MC error codes	41
UAC reserve controller CB (6xx) and EC (8xx) error codes reported by the RCD	45
UAC reserve controller MC error codes reported by the RCD	49
TDC sequence programs	53
6 System administration	1
Configuration management	2

TDC box configuration monitoring	5
TDC box configuration backup and restore	9
Troubleshooting	11
System monitoring points	13
7 Control Language reference	1
Configuration tasks	3
CL acronym libraries	3
Process module points	3
C Link addresses	5
Logic blocks	5
Program development	6
CL software environment	6
CL program directories	6
Creating CL programs	7
CL sequence messages	7
Compiling CL programs	9
Compile time errors	9
Downloading sequence programs	11
Troubleshooting sequence programs	11
Sequence program run time errors	12
Uploading sequence programs	12
Comparison with the TPS CL/MC compiler	13
Control Language basics	16
Rules and elements	17
Data types	28
Variables and declarations	32
Expressions and conditions	36
Statements	43
Program statement definitions	44
Embedded compiler directives	66
Structures	67
Sequence program definition	68
Abnormal condition handlers	72
Restart routines	75
Accessing parameters	76
User-written subroutines	87
Built-in functions and subroutines	89
Syntax summary	91
Syntax diagram summary	92
Notation used for syntax production rules	102
Production rules	103

Getting started

This document describes how to connect and configure the Experion system with a TDC 3000 Data Hiway.

For information about:	Go to:
Referenced documentation	page 2
Glossary of terms	page 3

Referenced documentation

Refer to the following documents for Experion-related information:

- *Software Change Notice* – for the latest enhancements and known problems
- *Planning Guide* – for system overview and planning information
- *Installation Guide* – for the system requirements and software installation procedures
- *Quick Builder Reference*—for the configuration of points, controllers, stations
- *Configuration Guide* – for the system configuration procedures
- *Operators Guide* – for the system operating procedures
- Reference Guides – for information on specific product options

These documents are available in the online documentation set known as Knowledge Builder.

Refer to the following documents for TDC Data Hiway related information:

- TDC 3000 Basic Product Manual Set
- SY-27-01—*SOPL User's Manual*
- SY-09-02—*Systems Interface Manual*
- PI-02-320— *Process Interface Unit General Description*
- SY-03-06—*20 HIM System Interface Manual*

Other reference documents:

- *Data Hiway Ethernet Bridge Installation Guide* – supplied with the DHEB hardware
- *Data Hiway Bridge User Guide* – supplied with the DHB hardware
- Third party documentation – for example, network and computer equipment documentation

Glossary of terms

Term	Description
AI	Analog Input.
AMC	Advanced Multifunction Controller, TDC Data Hiway device.
AO	Analog Output.
BOS	Basic Operator Station, Honeywell TDC Data Hiway device.
Box	Any TDC Data Hiway process interface device.
C Link	Local Communication Link used by the MC.
CB	Basic Controller, Honeywell TDC Data Hiway device.
CI	Counter Input.
CL	Abbreviation for CL/MC.
CL/MC	MC programming language supported by TPS systems and PlantScape systems.
DHB	Data Hiway Bridge, current serial TDC Data Hiway interface device.
DHEB	Data Hiway Ethernet Bridge, new TDC Hiway interface device, formerly DHN.
DHP	Data Hiway Port, Honeywell TDC Data Hiway device.
DI	Digital Input.
DO	Digital Output.
EC	Extended Controller, Honeywell TDC Data Hiway device.
EOS	Enhanced Operator Station, Honeywell TDC Data Hiway device.
Expanded PIU	Requires two Data Hiway Addresses rather than one address.
Experion	Honeywell control system product range, formerly PlantScape
HG	Hiway Gateway, Honeywell TDC Data Hiway interface device.
HIM	Honeywell Interface Module, Honeywell TDC Data Hiway device.
HLPIU	High Level Process Interface Unit, Honeywell TDC Data Hiway device.
HTD	Hiway Traffic Director, Honeywell TDC Data Hiway device.
LCS	Logic Control System, Honeywell PLC.
LEPIU	Low Energy Process Interface Unit, Honeywell TDC Data Hiway device.
LLPIU	Low Level Process Interface Unit, Honeywell TDC Data Hiway device.

Term	Description
MC	General abbreviation for AMC or MFC, Honeywell TDC Data Hiway device.
MFC	Multifunction Controller, Honeywell TDC Data Hiway device.
PAD	Preferred Access Device, for example, HG, DHB, EOS, BOS etc.
PAE	Preferred Access Expander, Honeywell TDC Data Hiway device.
PCFA	Point Card File Assembly.
PIU	Process Interface Unit, Honeywell TDC Data Hiway device.
PLC	Programmable Logic Controller.
Slot	TDC term applied to an I/O card slot or regulatory control loop.
SOPL	MC programming language supported by the TDC Enhanced Operator Station.
Station	Operator Station.
Subslot	TDC term applied to a single input on an I/O card.
TDC	Total Distributed Control, Honeywell product range.
TPS	TotalPlant Solution, Honeywell products and services.
UAC	Uninterruptable Automated Control, the TDC term for controller redundancy.

System comparison

2

This chapter describes the differences between the Experion system and the TDC Data Hiway system. Additional information is available in the documents listed in the section “Referenced documentation” on page 2.

For information about:	Go to:
An overview of system differences	page 2
TDC box support	page 5
Operational considerations	page 9

Overview

There are many differences between the Honeywell Experion system and the old TDC Data Hiway system that result from the different market requirements and trends over several decades. The TDC Data Hiway system is a Distributed Control System based on Honeywell proprietary standards. The Experion system was designed to handle a much wider range of requirements and is based on open computing standards. The following table provides an overview of the main differences between the two systems.

Topic	Experion System	TDC 3000 BOS/EOS
System size	Multiple server architectures are supported for enterprise and plant wide systems.	Single Data Hiway.
Communication standards	Open standards support for third party systems and data exchange between different systems.	Proprietary Data Hiway network.
System architecture	Redundant architectures are used to provide high system availability.	The operator stations are connected directly to the Data Hiway in parallel.
Database size	Up to 65,000 points per server.	4000 points per BOS, 1500 points per EOS
Database management	Automatic database synchronization between redundant servers.	Each operator station has an independent database.
Controller communications	Supports a wide range of Honeywell and third party controllers. Regularly scans all TDC Data Hiway based data.	Only supports the Data Hiway devices. Scans the data on an “as needed” basis depending on station assignments and the current display.
Displayed values	Some values are displayed in engineering units instead of percent. Differences in the data types can cause minor differences in the values displayed on each system.	Not applicable.
Alarming	Determined by server. Some of the alarm limits are tracked between the server and box.	Determined by the box.
Alarm displays	Alarms are reported by plant areas, areas are assigned to operators and operator stations. All alarm states are displayed for example, PVHIHI plus PVHI.	Alarms are reported by groups, the displays only show the most significant alarm for example, PVHIHI.

Topic	Experion System	TDC 3000 BOS/EOS
Alarm annunciation	Alarm indicator LEDs and klaxon outputs are supported via the available TDC keyboard outputs, or controller outputs.	Supported by the operator station.
Periodic history data	Provides superior history collection, data archiving and trend display facilities. History is available system-wide to all users.	Supports TDC controller and local operator station history collection, both of which are very limited.
Event history data	Provides event history collection, data archiving and display facilities.	Not supported.
Sequence of events	Supports HLPIU sequence of event data.	Not supported.
Data Hiway recording	Supported using point algorithms or simple server scripting.	Supported.
Custom displays	Superior custom display facilities using HMIWeb Technology.	Not supported by BOS. The EOS supports old style character based displays.
EOS calculation points	Supported using point algorithms or simple server scripting.	The EOS supports Calculation Points using subset of the SOPL language.
EOS sequence libraries	Provides 20 CL/MC Acronym Libraries.	EOS supports 4 Sequence Libraries.
EOS sequence master	The EOS or LCN Sequence Mastership assignments disable the sequence message handling on the Experion system. See “CL sequence messages” on page 7 for details.	EOS Stations can be assigned to be the Sequence Master or Slave for each MC.
EOS sequence displays	Requires custom displays.	EOS provides Unit Summary and Unit Detail displays.
EOS batch control	Batch control requires TotalPlant Batch or Recipe Manager. A simple Batch report is available.	Optional EOS disk set.
TDC group overview display	Use custom displays.	Supported.
TDC Data Hiway commands	The STR command must be used in place of the BOS/EOS INIT command, “TDC Data Hiway commands” on page 4 for details.	Supported.

Topic	Experion System	TDC 3000 BOS/EOS
TDC box types	Supports all box types except for the old Analog Unit box type. Some restrictions apply, “TDC box support” on page 5 for details.	The EOS supports the Multifunction Controller but not Extended Controller. The BOS supports the Extended Controller but not Multifunction Controller. PIU/DHP 32 bit counter inputs are not supported HLPIU Sequence of Events data is not supported
TDC box configuration	Windows based configuration via drop-down multi-choice lists.	Configuration via numeric codes.
TDC I/O allocations	Dual digital inputs and outputs must be assigned to consecutive subslots.	Supports PIU/DHP/HIM dual inputs and outputs with nonconsecutive subslots.

TDC box support



Note

The Data Hiway Box revision levels must be as stated in the TDC 3000 LCN R400 Hiway Gateway manual or later.

The following TDC 3000 boxes are supported with some limitations:

Controller Type	Acronym	Description
Basic Controller	CB	The Basic Controller handles continuous Input/Output operations. It accommodates I/O for eight modulating slots, plus eight analog inputs. It supports Rev 1C (or later). The box trend and average values are not supported. Use the Experion history collection and trending functions.
Extended Controller	EC	The Extended Controller handles continuous Input/Output and logical operations. It provides 16 modulating slots, 16 status outputs (8 SOA, 8 SOB), and 16 optional status inputs. A point's SP value is not linearized according to the regulatory slot Y input configuration setting. When linearization is configured on the X input, a PV algorithm is automatically added to the associated point and linearization is applied to the point's PV and SP values. This functionality is not supported for the Y input linearization setting. If linearization is configured on the X input the same setting should be used on the Y input. Points with an SP parameter but no PV parameter cannot be linearized. The box trend and average values are not supported. Use the Experion history collection and trending functions.

Controller Type	Acronym	Description
Multifunction Controller Advanced Multifunction Controller	MFC AMC	The Multifunction Controller (MFC) and Advanced Multifunction Controller (AMC) handle continuous Input/Output operations, logical operations, and sequential control for batch operations. Each MFC and AMC has the capacity to control 16 modulating slots and can handle: up to 32 analog inputs, up to 72 analog outputs, up to 256 digital inputs, and up to 64 counter inputs. The I/O Slots are numbered 17 to 32 instead of 1 to 16. Experion utilizes CL/MC sequence programming language instead of SOPL. Analog Input accumulation, Counter Input and Timer Start/Stop/Reset control is supported via the controller configuration displays using the supervisor login. CL/MC control schemes can be used to control these states from the point details displays if required. The Counter Inputs and Timers use 24 bits but only 16 bits are supported. CL/MC control schemes can be used to display the full range if required. Digital Input override can be enabled but the input state selection is not supported. Digital Input/Output Feedback Alarms can be configured but they are not alarmed on Experion. Use the Status Point Control Failure alarms or use CL/MC control schemes to display the Feedback alarm state if required. The box trend and average values are not supported. Use the Experion history collection and trending functions.

Controller Type	Acronym	Description
Process Interface Units	HLPIU LLPIU LEPIU	PIUs are used when a large number of process variables need to be monitored or controlled. PIUs come in low-level (LLPIU), high-level (HLPIU), and low-energy (LEPIU) versions. A normal PIU has one Data Hiway address. Expanded PIUs have two Data Hiway addresses. Only the lower 16 bits of the 32 bit counters are supported. Use the Accumulator Point type to accumulate the lower 16 bits if required. Dual digital inputs and outputs must be assigned to consecutive subslots. Digital Input/Output Feedback Alarms can be configured but they are not alarmed on Experion. Use the Status Point Control Failure alarms if required. HLPIU momentary digital outputs are not supported. Use the HMIWeb display scripting to reproduce this functionality if required.
Data Hiway Port LCS 620 Hiway Interface Module	DHP HIM	The Data Hiway Port (DHP) provides a generic interface for non-Honeywell devices, for example: programmable controllers, analyzers, tank gauging systems, machinery monitoring systems, emergency shutdown systems, data acquisition systems, and compressor control systems. The 620 LCS programmable controller can optionally interface directly to the Data Hiway using a Hiway Interface Module (HIM) that plugs into the processor rack of the controller. One 620 LCS HIM will appear as one to four Data Hiway Ports. The LCS 620 programmable controller offers capabilities ranging from relay ladder logic to advanced function capabilities. A wide set of Input/Output modules is available. The DHP and HIM are only supported as an expanded device with two Data Hiway addresses. Dual digital inputs and outputs must be assigned to consecutive subslots. Only the lower 16 bits of the 32 bit counters are supported. Use the Accumulator Point type to accumulate the lower 16 bits if required.

Controller Type	Acronym	Description
Reserve Basic Controller Director	RCD	The Reserve Basic Controller Director, Reserve Extended Controller Director and Reserve Multifunction Controller Director are used to monitor the status of the primary and backup controllers when redundant controllers are configured. They are also used to switch control from the primary controller to the backup, and back again to the primary if required. The reserve controller and reserve controller director control functions are supported but the status indications are limited.
Reserve Multifunction Controller Director	RMCD	
Reserve Extended Controller Director	RECD	

Operational considerations

This section identifies some differences that may impact the existing TDC Data Hiway operating procedures:

- The BOS/EOS Data Hiway status indications include the communication status and the box status. These states are normally reported separately on the Experion system displays.
- The reserve controller and reserve controller director status indications are limited.
- Box errors generate a communication alarm but do not generate alarms on the associated points.
- The point detail displays do not include the box status alarm indicator.
- The output signals cannot be controlled if the box is in the reset state or has failed.
- All alarm states are displayed instead of a single alarm state for example, PVHIHI plus PVHI instead of PVHIHI.
- The analog point output limits are applied in all modes, including manual mode; “TDC point property settings” on page 16 for details.
- Some values are displayed and entered in engineering units instead of percent.
- After configuring the system, the server database should be regarded as the master database and Quick Builder (or point build files) should be used as a backup data source. For more information about data sources, see “Configuration management” on page 2.
- If the TDC controller resident configuration is changed from another system (for example, LCN, BOS, EOS or another computer system) connected to the same Data Hiway, the controller configuration must be repeated or uploaded to the server to update the information on the Station displays.

Data Hiway interfaces

3

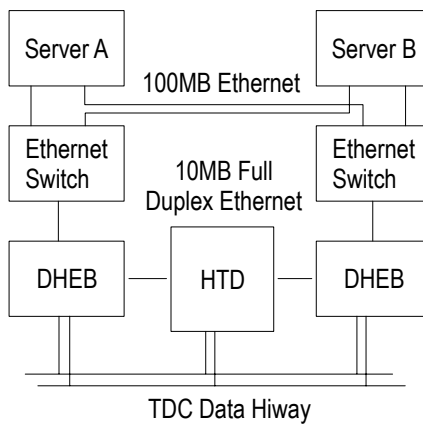
This chapter provides an introduction to the possible system architectures and the TDC Data Hiway interface connections. Additional information is available in the documents listed in the section “Referenced documentation” on page 2.

For information about:	Go to:
Data Hiway Ethernet Bridge	page 2
Data Hiway Bridge	page 10

Data Hiway Ethernet Bridge

Data Hiway Ethernet Bridge architectures

The Data Hiway Ethernet Bridge (DHEB) provides a high performance Ethernet interface to the TDC Data Hiway subsystem. The following diagram provides an overview of the recommended system architecture for a fully redundant single Data Hiway system using two Data Hiway Ethernet Bridges, other architectures are possible, please contact Honeywell for details.



The Experion servers can communicate via either Ethernet switch, thereby providing a redundant path to the TDC Data Hiway. The Experion servers should be configured for a redundant Channel. If the primary server path to one of the Data Hiway fails, for example due to a network, switch, or DHEB failure, the communications will failover to the backup path automatically. During normal operations, the server performs a diagnostic scan via the backup path every 60 seconds. If both primary server paths to the Data Hiway fail, the server failover must be performed manually, but if the server fails, the switchover is performed automatically.



Note

The switch ports connected to the DHEBs must be configured for 10 MB Full Duplex communications.

The DHEB does not support Data Hiway cable switching (unlike the DHB) so a Hiway Traffic Director (HTD) is always required with a DHEB. The HTD should be the later HT20 type that uses the round robin preferred access scheme if there are other Preferred Access devices (for example, computer, HG, EOS or BOS) connected to the Data Hiway.

Fault Tolerant Ethernet architecture

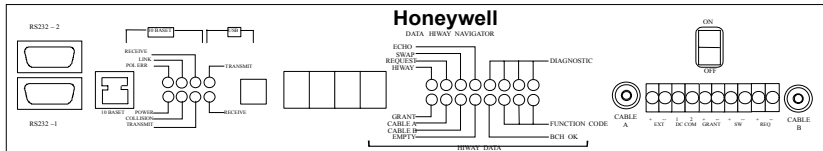
The DHEB does not support dual Ethernet connections but it can be connected to an Fault Tolerant Ethernet (FTE) network. The same network architecture is used for FTE except that a crossover connection is required between the switches and each Station has two Ethernet connections, one to each Ethernet switch.

The DHEB is sensitive to other network traffic (this can cause DHEB connection timeouts), so the Ethernet switch ports connected to the DHEBs should be configured to block the FTE Multicast traffic.

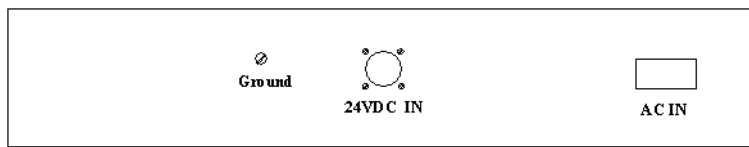
For more information about FTE, see the *Fault Tolerant Ethernet Overview and Implementation Guide* and the *Fault Tolerant Ethernet Installation and Service Guide* section in Knowledge Builder.

DHEB hardware

The DHEB is designed for installation in a 19 inch equipment rack. The front panel contains the USB connector, 10BaseT connector, 2 RS232 connectors, status LEDs, Hiway Traffic Director connections, Data Hiway cable connections, power switch and HEX displays.



The rear panel contains the 24VDC, AC, and Ground connections. The AC input range is 85-264 volts.



The CD included with the DHEB hardware contains the *DHEB Installation Guide*, firmware and server software components. The following sections provide a summary of the DHEB setup and software installation procedures.

DHEB hardware setup

The DHEBs must be configured with three settings:

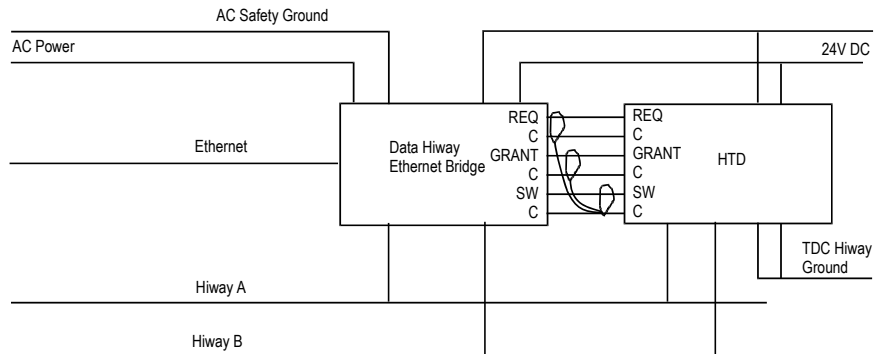
- 1 A Hiway Box Number that must be set to an unused Data Hiway address between 1 and 3. When multiple DHEBs are connected to the same Data Hiway they must have the same Hiway Box Number. This setting is set via internal switch pack 1 (S1) as described in the *DHEB Installation Guide*.
- 2 A Hiway Number (between 1 and 14) that uniquely identifies the DHEB to the DEHB interface software installed on the Experion servers. This setting is set via internal switch pack 3 (S3) as described in the *DHEB Installation Guide*.
- 3 An IP Address that uniquely identifies the DHEB to the DEHB interface software installed on the Experion servers. This setting is set via a serial port connection as described in the *DHEB Installation Guide*.

DHEB connections

See the *DHEB Installation Guide*.

DHEB grounding considerations

The DHEB must be used with a Hiway Traffic Director (HTD), the preferred access cable shields must be grounded at the HTD only as shown in the following figure.



The 24VDC common is connected to the zero volt connection located in the power socket at the rear of the DHEB. The 24V power to the DHEB is electrically isolated to 500v from the Data Hiway.

The AC safety ground should be connected to the ground stud located at the rear of the DHEB.

DHEB software installation

The software supplied with the DHEB hardware must be installed on the Experion server as follows:

- 1 Check that the hard drive has at least 120MB of unused space.
- 2 Check the user documentation on the CD for any last minute information.
- 3 Run the Setup .EXE file located on the CD to start the installation.
- 4 Use the default installation directory and follow the instructions given by the setup program.
- 5 If requested, reboot the computer to complete the installation.
- 6 For redundant Experion systems, repeat the software installation on the other server.

DHEB software configuration

After completing the software installation, start the Monitor program and configure the interface as follows:

- 1 Select **Start > Programs > DHHEB > DHEB MONITOR** to start the Monitor program.
- 2 Select **Configure > DHEB Interface**. The Interface Configuration dialog box opens.
- 3 Click the **Edit** button and enter the configuration details, where:
 - Hiway Number Corresponds with the DHEB switch settings, see the DHEB Set Up
 - Hiway Description with the recommended entry as **HIWAYnn** where **nn** is the same as the Hiway Number.
 - IP Address The address assigned to the DHEB, see the DHEB Set Up procedure

DHEB Configuration File: J:\PROGRAM\1\HONEYW\1\DHEB\bin\DHEBConfigDB.mdb

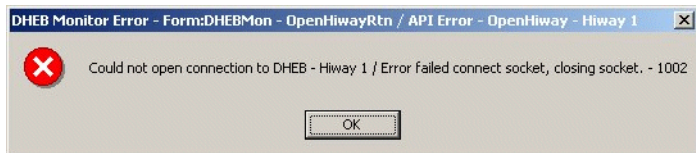
	Hiway Number	Hiway Description	IP Address
DHEB #: 1	1	HIWAY01	159.99.7.116
DHEB #: 2	2	HIWAY02	159.99.7.117
DHEB #: 3	0		
DHEB #: 4	0		
DHEB #: 5	0		
DHEB #: 6	0		
DHEB #: 7	0		
DHEB #: 8	0		
DHEB #: 9	0		
DHEB #: 10	0		
DHEB #: 11	0		
DHEB #: 12	0		
DHEB #: 13	0		
DHEB #: 14	0		

Initialize Edit Save Cancel

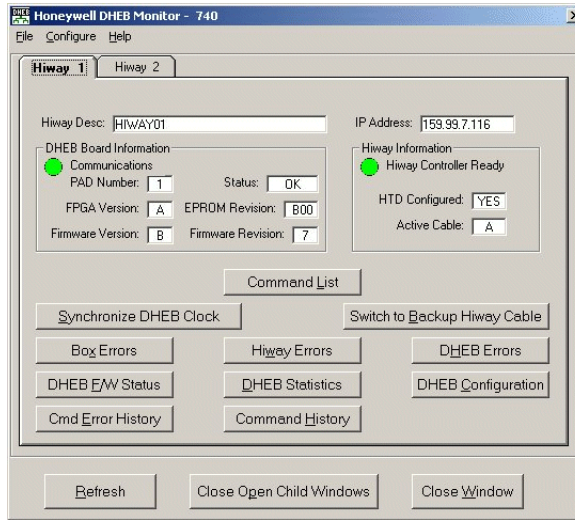
Enter "0" (zero) for Hiway Number to Disable the DHEB

4 Click **Save** to save the DHEB interface configuration.

The following error appears if the Monitor program cannot communicate with the DHEB.



The following dialog box is displayed if the communications are successful.



If this dialog box is not displayed see “DHEB troubleshooting” on page 9.



Caution

The DHEB Monitor program can also be used to read and write data to the online process connected boxes, which can cause box data corruptions. If you use the Monitor program to read/write to the online process connected boxes, you do so **AT YOUR OWN RISK!**

If the communications are successful, go to the Experion system configuration procedures (see “System configuration” on page 1).

DHEB troubleshooting

If communications are not successful, follow these steps:

- 1 Check the DHEB front panel Hex display, the display should show *hhbb* where *hh* is the Hiway Number and *bb* is the Hiway Box Address. See the *DHEB Installation Guide* for other display indications.
- 2 Select **Start > Programs > Accessories > Command Prompt** and type the command:

```
Ping <IP Address>
```

where <IP Address> is the IP Address assigned to the DHEB.

If there is no response from the DHEB check the network equipment, cabling and DHEB IP Address.

- 3 The DHEB can support up to 6 communication connections, the Experion servers use 4 connections to the DHEB and the DHEB Monitor program uses 1 connection so if there are other computers running the DHEB Monitor program or Experion servers trying to communicate with the DHEB, it is possible that the maximum number of connections has been exceeded so close any unnecessary connections then retry the communications via the Monitor program.

The following messages are listed in the Experion server log file when all connections to the DEHB are in use:

```
tdcscn.exe:dhnio.cpp,v:362: DHEB error 11: All connections in use
```

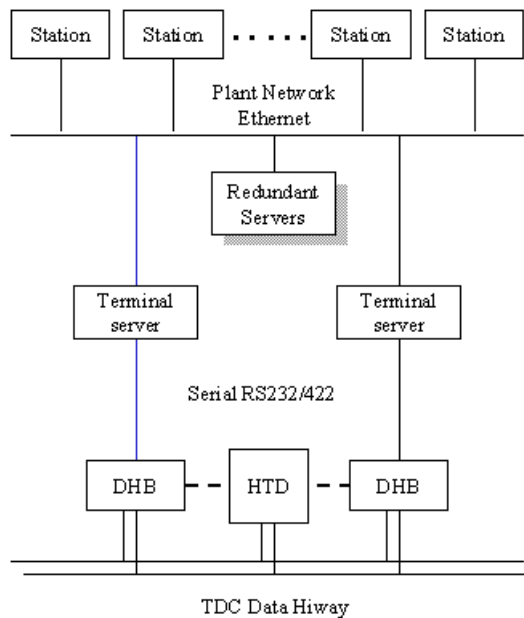
```
tdcscn.exe:dhnio.cpp,v:497: DHEB error 1001: Socket error/timeout, socket has been closed
```

To check the server log file go to the Primary Server and select **Start > Programs > Honeywell Experion PKS > Server > Diagnostic Tools > Experion PKS Server Log**.

Data Hiway Bridge

Data Hiway Bridge architectures

The Data Hiway Bridge (DHB) provides a serial interface to the TDC Data Hiway subsystem. The following diagram provides an overview of the recommended system architecture for a fully redundant single Data Hiway system using two Data Hiway Bridges, other architectures are possible, please contact Honeywell for details.

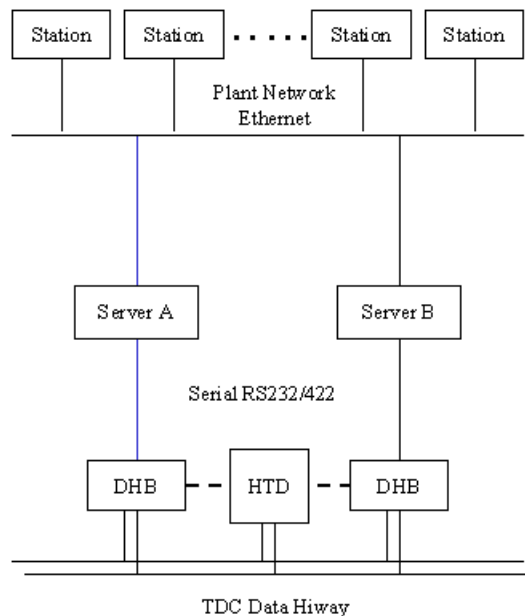


The terminal servers provide the interface between the Ethernet network and the serial communication links to the DHBs. The Experion servers can communicate via either terminal server, thereby providing a redundant path to the TDC Data Hiway. The Experion system should be configured for a redundant Channel. If the primary server path to one of the DHBs fails, the communications will fail over to the back up path automatically. During normal operations, the server will perform a diagnostic scan via the backup path every 60 seconds.

See to the *Installation Guide* for a list of qualified terminal servers and detailed set up instructions.

The Hiway Traffic Director is required if the maximum Data Hiway cable length will be exceeded, the maximum number of devices is exceeded or when there is more than one Preferred Access device (for example, computer, HG, EOS, BOS or DHB) connected to the Data Hiway. The HTD should be the later HT20 type that uses the round robin preferred access scheme if there are other Preferred Access devices connected to the Data Hiway.

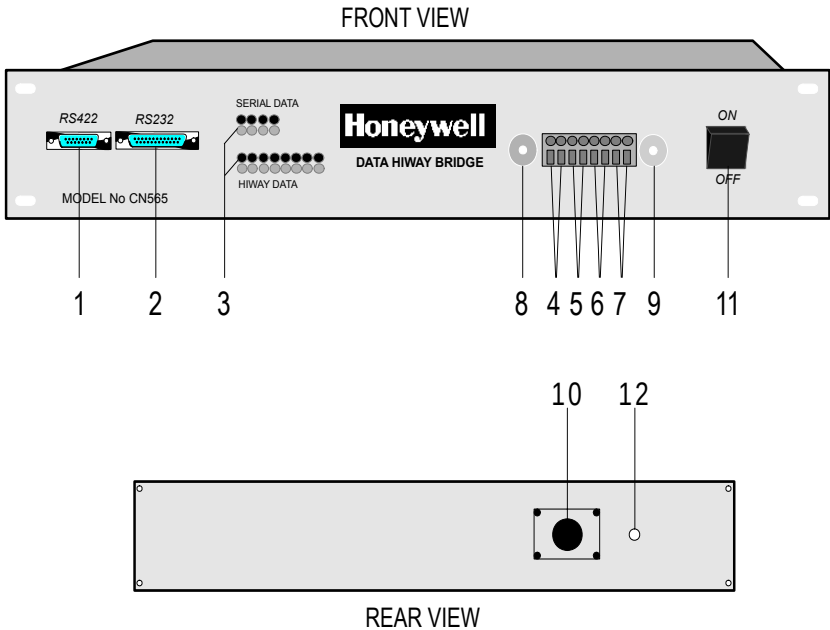
The following diagram provides an overview of an alternative redundant single Data Hiway system without terminal servers using two Data Hiway Bridges (DHB), other architectures are possible, please contact Honeywell for details.



Each Experion server has a single communication path to the TDC Data Hiway. The Experion system should be configured for a non-redundant Channel. If the primary server path to one of the DHBs fails, a manual server fail over must be performed in order to re-establish communications with the TDC Data Hiway.

DHB hardware

The DHB is designed for installation in a 19 inch equipment rack.



1	RS-422 socket	7	Grant Lines
2	RS-232 socket	8	Hiway cable A socket
3	Status LEDs	9	Hiway cable B socket
4	Alarm contacts	10	24 Vdc socket
5	Request Lines	11	Power on/off
6	Swap Lines	12	Earth connection point

DHB hardware setup

The DHB communications settings are set with switches on the main circuit board. The switches set the:

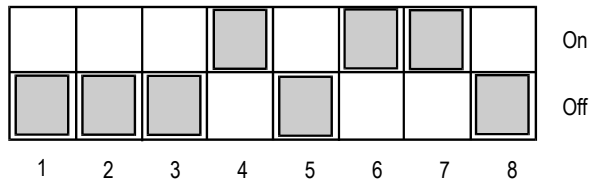
- Baud rate (baud rate should be fast for best performance, 19200 or higher)
- Parity (usually EVEN)
- Protocol (RS-232)
- Hiway Address (3, for example)

Record these settings because they are required when you configure the channel on the server.

Hiway Bridge switch settings

Switch bank S3

Figure 1 Settings for S3



Switch 1 is ON if RS-232 parity is to be disabled (the default is OFF).

Switch 2 is ON if Hiway retries are inhibited (the default is OFF). The recommended setting is ON, retries disabled.

Switch 3 is ON if a Hiway Traffic Director is present on the Data Hiway (the default is OFF).

Switches 4 to 7 give the RS-232 and RS-422 baud rates as follows. Switch 8 is ON if Hiway writes are to be inhibited (the default is OFF).



Note

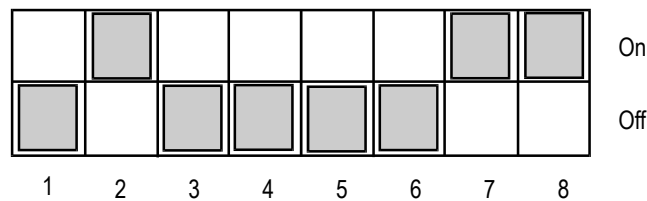
The following settings depend on the IC used in position U6 on the processor board BM403. The default baud rate is 9600.

SW4 to SW7 (1 = ON)	Rate if U6 is 68661PB or SCN26661BC	Rate if U6 is 68661PC
0000	45.5	50
1000	50	75
0100	75	110
1100	110	134.5
0010	134.5	150
1010	150	300
0110	300	600

SW4 to SW7 (1 = ON)	Rate if U6 is 68661PB or SCN26661BC	Rate if U6 is 68661PC
1110	600	1200
0001	1200	1800
1001	1800	2000
0101	2000	2400
1101	2400	3600
0011	4800	4800
1011	9600	7200
0111	19200	9600
1111	38400	19200

Switch bank S2

Figure 2 Settings for S2



Switches 1 to 6 contain the binary encoded Data Hiway address of the Bridge from 0 to 63 (the default is 2).

Switch 7 is ON for even parity; the default is ON.

Switch 8 is ON for RS-232 operation and OFF for RS-422 operation; the default is ON.

SW1 to SW6 (1 = ON)	Data Hiway Address
100000	1
010000	2
110000	3

**Note**

A Hiway Box Number that must be set to an unused Data Hiway address between 1 and 3. When multiple DHBs are connected to the same Data Hiway they must have the same Hiway Box Number.

DHB connections

Server to Data Hiway Bridge cable

Figure 3 RS-232 Wiring for server with DB 9 connector

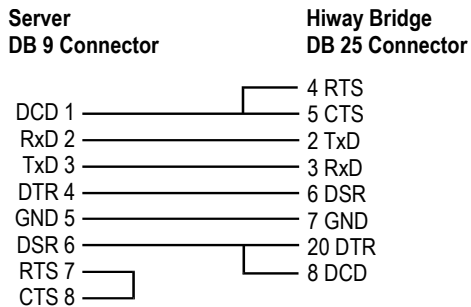


Figure 4 RS-232 Wiring for server with DB 25 connector

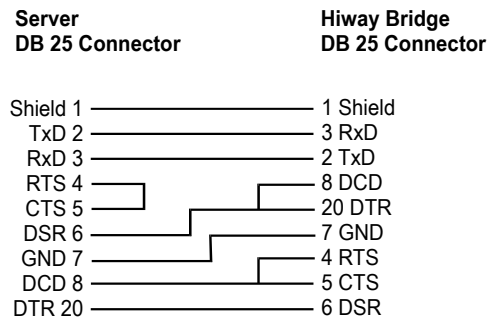
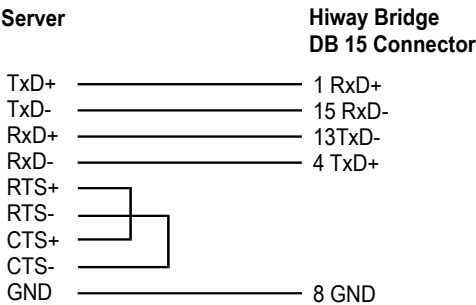


Figure 5 RS-422 wiring



Data Hiway Bridge preferred access cable

If a Hiway Traffic Director (HTD) is present, connect the six preferred access lines to the Hiway Traffic Director as follows. The Data Hiway Bridge may be placed a maximum of 1500m from the HTD. Shielded twisted pair cable should be used.

Data Hiway Bridge	HTD
Request	Request +
Request C	Request –
Grant	Grant +
Grant C	Grant –
Swap	Swap +
Swap C	Swap –

If no Hiway Traffic Director is present, connect the preferred access lines for self grant as follows.

Data Hiway Bridge	Data Hiway Bridge
Request	Grant
Request C	Grant C
Swap	No connection
Swap C	No connection

Data Hiway Bridge Hiway connection

To connect the DHB to the TDC Data Hiway carry out the following steps:

- 1 Connect a 75ohm BNC T-piece to the Data Hiway Port A socket.
- 2 Connect a 75ohm BNC T-piece to the Data Hiway Port B socket.
- 3 Connect the Data Hiway A cable to the Port A T-piece.
- 4 Connect the Data Hiway B cable to Port B T-piece.
- 5 If the DHB is at the end of the cable, connect a 75ohm BNC terminator to both T-pieces; otherwise, connect continuation cables to both T-pieces.

Data Hiway Bridge power connection

Connect the power cable to a 24Vdc power supply as follows.

Wire	PSU
Red	24V
Black	0V, common



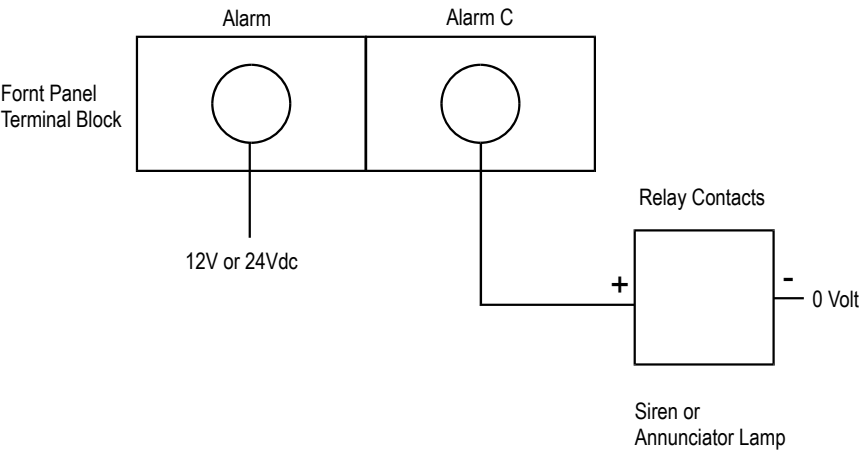
Note

0V is internally connected to the Data Hiway Bridge chassis. There is 500Vdc isolation between the 24Vdc supply and the Data Hiway Bridge.

Data Hiway Bridge alarm connection

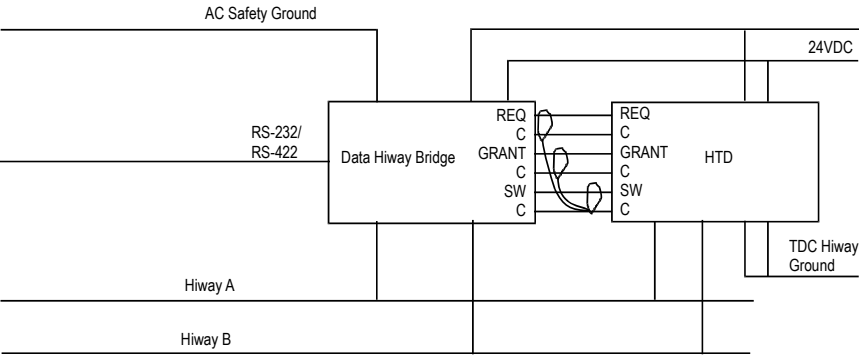
The normally open (0.25A) relay contact is closed if an internal fault relating to Hiway operation is detected. Do not apply over 32V to these contacts.

Figure 6 Recommended External Connection To Alarms

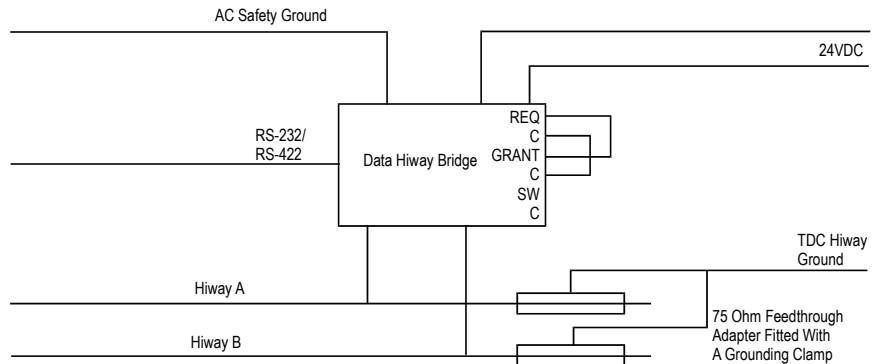


DHB grounding considerations

When the system has a Data Hiway Bridge as well as a Hiway Traffic Director, the preferred access cable shields must be grounded only at the HTD, as shown in the following figure.



If the system has a Data Hiway Bridge and no HTD, the Data Hiway cable shield must be grounded only at the Data Hiway Bridge, as shown in the following figure.



Grounding of the Hiway cable shields at the Data Hiway Bridge is achieved by two methods:

- For Hiways terminated at the Data Hiway Bridge, a pair of 75ohm BNC terminating plugs fitted with a grounding strap should be used. The grounding strap should be connected to the Hiway ground.
- For Hiways not terminated at the Hiway Bridge, a BNC feed-through adapter fitted with a grounding clamp should be installed in each Hiway at the Data Hiway Bridge. The grounding clamps should be connected to the Hiway ground.

The connection to the Hiway ground in either of the above methods should be insulated. All other exposed metal parts of the BNC connectors used on the Hiway should be protected with an insulated connector cover.

The 24VDC common is connected to the zero volt connection located in the power socket at the rear of the DHB. The 24V power to the DHEB is electrically isolated to 500v from the Data Hiway.

The AC safety ground should be connected to the ground stud located at the rear of the DHB.

DHB troubleshooting

The server configuration must be performed before the communications can be tested, “System configuration” on page 1 for details.

System configuration

4

This chapter describes how to configure the Experion system using Quick Builder. Additional information is available in the documents listed in the section “Referenced documentation” on page 2.

After configuring the system, the server database should be regarded as the master database and Quick Builder (or point build files) should be used as a backup data source. For more information about data sources, see “Configuration management” on page 2.

Before configuring the TDC Data Hiway items it is necessary to create a Quick Builder project with the TDC option enabled. For instructions on creating a Quick Builder project, configuring the server properties, adding Stations and Printers, see the Quick Builder reference. Then define the TDC Channels as described in the following sections.

For information about:	Go to:
TDC channels	page 2
TDC controllers	page 7
TDC Points	page 11
Additional tasks	page 39

Defining TDC channels

The Channel configuration defines the communication links between the server and the TDC Data Hiway.

To define a channel:

- 1 Click  to display the Add Items dialog.
- 2 Select `channel` as the item and `TDC` as the type.
- 3 Click **OK** to add the new item.
- 4 Click the **Main** tab and configure the properties. See “Channel Main tab settings” on page 2.
- 5 Click the **Port** tab and select the Port Type as follows:
 - `LANVendor` for a DHEB Channel. See “Channel LANVendor port settings” on page 4.
 - `serial` for DHB Channel with a direct connection to the server. See “Channel serial port settings” on page 4.
 - `Terminal Server` for DHB Channel with a terminal server connection to the server. See “Channel terminal server port settings” on page 5.
- 6 Click the **Redundant Port** tab and select the Port Type as described above if the Channel is redundant or Port Type of `none` if the Channel is non-redundant.
- 7 Click  to download the Channel definitions to the server.
- 8 Configure the Channels as described in “Channel configuration” on page 6.



Note

The Experion server application must be restarted if the Channel Port Type is changed for an existing Channel, for example, from Serial to Terminal Server or LANVendor.

Channel Main tab settings

The Main tab defines the basic properties for a channel.

Property	Description
Name	A unique name, for example, <code>HIWAY01</code> using a maximum of 10 alphanumeric characters (no spaces, underscores or double quotes).

Property	Description
Description (optional)	A description, for example, the process name, using a maximum of 30 alphanumeric characters.
Marginal Alarm Limit	The communications barometer alarm limit at which the Channel is declared to be marginal. When this limit is reached, a high priority alarm is generated. The Channel barometer monitors the total number of communication requests and the number of errors between the server and the DHEB or DHB. Redundant Channels have two barometer values, one for Link A and one for Link B. The barometer values are incremented by two or more, depending on the error and decremented for each good request. If the Channel is redundant, set the Marginal Limit to 3 and the Fail Limit to 5 for a fast fail over to the redundant link. If the communications are noisy these limits can be increased.
Fail Alarm Limit	The communications barometer alarm limit at which the Channel is declared to have failed. When this limit is reached, an urgent alarm is generated and the communications switch to the backup link if the Channel is redundant.
Connect Timeout	The time, in seconds, the server waits for a successful connection to the DHB or DHEB. The normal setting is 10 seconds.
Read Timeout	The time, in seconds, the server attempts to read data from the DHB or DHEB before giving up. Set this value to 5 seconds when using the DHEB. The default setting is 2 seconds.
Item Number	A unique Channel number for example, CHN01. The number must be between 1 and the maximum number of channels allowed for your system (see the Sizing tab for the Project item for the maximum number of Channels).

TDC channel port properties

The Port tab defines the communication-related properties for a channel. The properties vary according to the selected **Port Type**. Select the port type as follows:

- **LANVendor** for a DHEB Channel. See “Channel LANVendor port settings” on page 4.
- **Serial** for DHB Channel with a direct connection to the server. See “Channel serial port settings” on page 4.
- **Terminal Server** for DHB Channel with a terminal server connection to the server. See “Channel terminal server port settings” on page 5.

Channel LANVendor port settings



Note

For Redundant Channels, the Port Names are different for example, HIWAY01 and HIWAY02.

Property	Description
Port Name	This entry must have the format HIWAY nn where nn is the Hiway Number. See “DHEB software configuration” on page 6.

Channel serial port settings



Note

For Redundant Channels, the Port and Redundant Port settings are normally identical

Property	Description
Serial Port Name	The device name of the serial port for example, COM1
Baud Rate	The same value as the DHB internal switch setting, use the fastest setting supported by the DHB. See “Hiway Bridge switch settings” on page 13.
Number of Data Bits	The same value as the DHB internal switch setting. The default is 8.

Property	Description
Stop Bits	The same value as the DHB internal switch setting. The default is 1.
Parity	The same value as the DHB internal switch setting. The default is NONE.
Checksum	Not applicable, leave as NONE.
XON/XOFF	Not applicable, leave as NONE.
RS-232 settings	Not applicable.
RS-485 settings	Not applicable.

Channel terminal server port settings

See the *Installation Guide* for a list of qualified terminal servers and detailed set up instructions.



Note

For Redundant Channels using two terminal servers the TCP Host Names are different. If a single terminal server is used the TCP Port Numbers are different.

Property	Description
Terminal Server TCP Host Name	Enter the terminal server IP address or TCP host name used by the terminal server.
Terminal Server TCP Port No	The internal terminal server port number to which the DHB is connected for example, 8001.
Idle Timeout	The time, in seconds, the channel waits for a successful connection to the terminal server before closing the connection. The default is 180 seconds, a value of 0 indicates that the connection is never closed.
Checksum	Not applicable, leave as NONE.

TDC channel redundant port properties

The Redundant Port tab defines the communication-related properties for a redundant channel, which can be either:

- `LANVendor`. See “Channel LANVendor port settings” on page 4.
- `Serial`. See “Channel serial port settings” on page 4.
- `TerminalServer`. See “Channel terminal server port settings” on page 5.

Channel configuration

Configure TDC channels using the following procedure:

- 1 Select **Start > Programs > Honeywell Experion PKS > Server > Station** to start Station.
- 2 Change the Station security level to **mng**.
- 3 Select **Configure > System Hardware > Controller Interface > Channels** to call up the Channel Configuration Summary display.
- 4 Select the **Show redundant link status** check box if a redundant channel is being used.
- 5 Select each channel in turn and configure as follows:

Property	Description
Upload permitted	If selected, the TDC Box configuration can be uploaded (saved) to the server.
Download permitted	If selected, the TDC Box configuration can be downloaded (restored) from the server.
Refresh enabled for computer mode timeout	If selected, the server refreshes the Time Out Gates within the TDC controllers to keep the loops in computer mode. This enables the TDC controllers to know if communications with the server have been lost which causes the loops to revert to a backup control mode.
Hiway owner	If selected, the server is notified of alarms and changes that occur within the TDC boxes. Only one system connected to the Data Hiway should be set as Hiway Owner. Setting the server as Hiway Owner increases the communications load, see “Exception scanning” on page 12 for further details.
PIU/DHP significant change reporting	If selected, the server is notified of significant changes that occur in analog inputs within PIU, DHP and HIM boxes. This enables analog values to be scanned by exception rather than periodically. Note that you must also select the Hiway Owner check box for this function to work, see “Exception scanning” on page 12 for further details.

Defining TDC controllers

The Controller configuration defines the boxes on the TDC Data Hiway.

To define a controller:

- 1 Click  to display the Add Items dialog.
- 2 Select `Controller` as the item and `TDC` as the type.
- 3 Click **OK** to add the new item
- 4 Click the Main tab and configure the properties. See “Controller Main tab settings” on page 7.
- 5 Click  to download the Controller definitions to the server.
- 6 Test the communications as described in the section “Testing the communications” on page 9.
- 7 Define the TDC I/O Slots as described in “TDC I/O slot definitions” on page 9.

Controller Main tab settings

The Main tab defines the basic properties for a controller.

Property	Description
Name	A unique name for example, <code>AMC<hh>bb</hh></code> where <i>hh</i> is the Hiway Number/Channel Number and <i>bb</i> is the Controller Number/Box Address, using a maximum of 10 alphanumeric characters (no spaces, underscores or double quotes).
Description (optional)	A description for example, a process reference or physical location, using a maximum of 30 alphanumeric characters.
Channel Name	Select the Channel Name used to communicate with the Controller.

Property	Description
Marginal Alarm Limit	The communications barometer alarm limit at which the Controller is declared to be marginal. When this limit is reached, a high priority alarm is generated. The Controller barometer monitors the total number of Data Hiway requests and the number of errors for the Controller. Each Controller has two barometer values, one for Hiway cable A and one for cable B. The barometer values are incremented by two or more, depending on the error and decremented for each good request. The default Marginal and Fail limits settings are 25 and 50 respectively. The Marginal and Fail limits should be set to 32 and 64 respectively to give a similar Hiway switching characteristic to that described in the <i>TDC Systems Interface Manual</i>.
Fail Alarm Limit	The communications barometer alarm limit at which the Controller is declared to have failed. When this limit is reached, an urgent alarm is generated and the communications are switched to the backup cable. If the Data Hiway interface is a DHB without an HTD, switching from cable A to cable B to cable A is possible.
Box Address (decimal)	The decimal address of the Controller on the TDC Data Hiway. This value must match the address set on the Hiway Box.
Box Type	Select the Hiway Box type.
Unexpanded PIU	For PIUs select this check box if the PIU is unexpanded, that is, it uses one Data Hiway address.
CL Acronym Library Number	This selection only applies to Multifunction Controllers (MFC) and Advanced Multifunction Controllers (AMC). If CL Sequence Programs are used, the Controllers can be allocated to a CL Acronym Library between 1 and 20. The words used in the CL programs are stored in these libraries when a CL/MC Sequence Program is compiled, see the CL Programming Language section for further details.
Item Number	A unique Controller number, for example, RTU005. Where possible, this number should match the Box Address because the Data Hiway commands are issued using the Controller number, not the box address. The number must be between 1 and the maximum number of controllers allowed for your system (see the Sizing tab for the Project item for the maximum number of Controllers).

Testing the communications

A diagnostic utility, **tdctst**, is included as part of the server software. This utility tests communications between the server and the TDC Data Hiway.

Before testing can commence, the following must have been completed:

- The Data Hiway Interface setup must be completed.
- The Data Hiway Interface must be powered up.
- All cables must be connected.
- All boxes must be powered up.
- All boxes must be connected to the Data Hiway.
- The Channel and Controller configuration data must have been downloaded to the server without any errors.

The **tdctst** utility indicates which boxes are connected to the Data Hiway. A correct listing of a box indicates that communications between the server and that box are good.

To run **tdctst**:

- 1 Open a Command Prompt window, and type:
tdctst
- 2 When prompted, type the channel number of the Data Hiway, followed by **find**. For example:
CHN01
find

A list of addresses and box types on the Hiway appears. Check that these correspond to the physical connections on the Data Hiway and to the server controller definitions.

If this is a new Hiway and the boxes have not been initialized, then **tdctst** cannot correctly identify the boxes. This is indicated by a trailing question mark (?).

Note that the Transmit and Receive lights on the Data Hiway Interface flash rapidly during the find process.

- 3 Type **q** to quit.
- 4 To test the redundant communication channel, use **CHN01D**.

TDC I/O slot definitions

The I/O slots associated with MCs, PIUs, DHPs and HIMs must be defined before the points using these slots can be successfully downloaded to the server. The I/O

slots can be defined off-line or uploaded from the TDC boxes if the server is connected to the TDC Data Hiway equipment.

To access the controller configuration displays:

- 1 Select **Start > Programs > Honeywell Experion PKS > Server > Station** to start Station.
- 2 Change the Station security level to **mng**.
- 3 Select **Configure > System Hardware > Controller Interface > Controllers** to call up the Controller Configuration Summary display.
- 4 Click the controller name to call up the Controller Status display.
- 5 Click **Details** to call up the Controller Configuration display.
- 6 Select the I/O slots (for MCs see slots 17 to 32) to view the TDC box configuration.

The information displayed on the left of the display is the server resident configuration. Red question marks to the right of server configuration indicate that the TDC box resident configuration cannot be accessed. Differences between the server configuration and the box resident configuration are indicated as red text. See the section “TDC box configuration monitoring” on page 5 for further information.

If the server is connected to the TDC Data Hiway equipment and the TDC box configuration is correct then click **Upload** to upload the box configuration to the server.

If the system is being prepared off-line without the TDC Data Hiway equipment, select the Board Type for each of the I/O slots to match the TDC hardware. The Board Type selections are the minimum configuration required to build the points, the box database can be fully defined off-line if required.

Refer to the *TDC BOS/EOS Configuration Form Instructions* and the controller Algorithm reference manuals for details of the TDC box configuration settings.

After defining the I/O slots or uploading the box configuration use the **hdwconfig** utility to save the server resident box configuration to a backup file, see the section “TDC box configuration backup and restore” on page 9.



Note

After restoring a configuration file it is sometimes necessary to call up I/O slot definition displays to synchronize the server configuration before the points associated with the I/O slots can be built successfully.

TDC points

Introduction

Scanning is the process by which the server requests data from the controllers. The TDC controllers support periodic scanning and exception scanning.

With periodic scanning, the data is read on a periodic basis as defined by the point configuration. For example, PV scan period of 5 seconds.

Periodic scanning is:

- Easier to implement
- Places a steady load on the system
- Can be more efficient than exception scanning when the data is changing rapidly for example, during plant upsets.

With exception scanning, the server reads special addresses in the controller every second. These addresses indicate whether the data has changed. When the server detects the change, the server performs a separate scan to read the data that has changed. When exception scanning is used the point parameters are configured with a long scan period, for example, PV scan period of 300 seconds.

Exception scanning is:

- Dependent on the Hiway Ownership selections, the box type, the box configuration and the box parameter, this makes it more difficult to implement. See “Exception scanning” on page 12 for details.
- Under steady state conditions the system load is lower than periodic scanning.
- The system load is unpredictable when the data is changing rapidly for example, during plant upsets.



Note

Both types of scanning can be used with a TDC controller.



Caution

The small scan packets generated by Exception scanning can severely limit the data throughput.

Exception scanning

If there is more than one control system connected to a TDC Data Hiway, only one of the systems should be the Hiway Owner. Sequence of Events (SOE) reporting from HLPIUs can only be performed by the Hiway Owner.

If a TDC Hiway Gateway or TDC Supervisory computer system is being used, the Hiway Owner, Refresh Computer Timeout and Significant Change Reporting selections on the Channel Configuration display should be disabled. If the Server is the only computer device on the Data Hiway these selections can be used to enable exception reporting.



Note

Periodic scanning must be used to monitor the control actions performed by the regulatory control slots because the SP, PV, OP and Mode changes performed by the control algorithms are not reported by exception.

Box type	Reporting technique	Notified exception	Server as owner	Server not owner
CB	Status word	Analog alarms	yes	yes
	Change flag	SP, OP, MD changes by other PADS	yes	no
EC	Status word	Analog alarms	yes	yes
	Change flag	SP, OP, MD changes by other PADS	yes	no
MFC/AMC	Circular list	Analog alarms	yes	yes
		SP, OP, MD changes by other PADS	yes	yes
		Digital inputs	yes	yes
		Flag changes	yes	yes
		Counter rollover	yes	yes
		Sequence messages	yes	yes
		Sequence state	no	no
PIU	Exception Reports	Notify status changes in Sequence of events	yes	no
		Analog alarms	yes	no

Box type	Reporting technique	Notified exception	Server as owner	Server not owner
		Analog significant change (insure every analog subslot significant change is configured)	yes (if report significant changes enabled)	no
		Counter rollover	yes	no
	Change flag	SP, OP, MD changes by other PADS	yes	no
DHP/HIM	Exception Reports	Digital input changes	yes	no
		Analog alarms	yes	no
		Analog significant change (insure every analog subslot significant change is configured)	yes (if report significant changes enabled)	no
		Counter rollover	yes	no
	Change flag	SP, OP, MD changes by other PADS	yes	no

Technique	Description
Status words	Status words are used by Basic Controllers and indicate the alarm state of each loop of each modulating slot. Should a slot enter an alarm condition (that is, PV HI, PV LO, DEV HI or DEV LO), a bit will be set in the status word. During its exception poll, the server system scans the status word in the Basic Controller. If a change since the previous poll is detected, the slot is checked to determine if a point parameter exists for it. If a point parameter does exist, the slot is scanned immediately and the point is processed. As server alarm limits track the box alarm limits, an alarm will be registered.

Technique	Description
Circular lists	A circular list is used by both MFC and AMC to notify the server of an alarm. At each exception poll the server will monitor the pointer of the circular list. If a change from the previous scan is detected, the list is further interrogated in order to determine the new entries. These new entries indicate slot number and alarm type. This information is used to initiate an immediate scan and process the point. For MFC and AMC, in addition to modulating slot alarm reporting, point card file assembly (PCFA) analog and digital input alarms are also reported. The alarm condition is as determined by the Hiway Box. For analog points, alarm limits must be tracked by the server. For digital points, the alarm state must match the state that is configured in the server. Changes of state of the first 64 flags of a Multifunction Controller are also reported by the circular list, as are counter rollovers and sequence slot changes. Analog alarms and counter rollovers will cause an immediate scan and point processing. Digital alarms and flag changes will be processed without a scan, as the circular list entry also contains the new state. Changes made by other preferred access devices (PADs) will be reported by way of the circular lists.
Change flags	If the server is the Hiway Box owner, other Preferred Access Devices (for example, BOS, EOS) will set change flags when they change SP, OP, and MD values, or tuning constants. As these changes are sent to the server, constant scanning of these values is not required. The following scanning strategy is recommended for modulating loops. SP. Use exception scanning unless the modulating loop can be placed in cascade mode, in which case use SP period. OP. OP period is required. MD. Use exception scanning (MD period is not required). Tuning Constants. Do not address. Use the Controller Configuration display on Station to view and change these values. If a value is required in a point record, use exception scanning.

Technique	Description
Exception Reports	The PIU, Data Hiway Port (DHP), and Hiway Interface Module (HIM) can be assigned an alarm reporting device to which they will send exception information. If the server is configured as the Hiway owner, it will be nominated as the alarm reporting device. To indicate that it has an exception to report, the PIU, DHP, or HIM will set a bit in a notification word. The server checks this location during the exception poll and, if the notification bit is set, an Exception Report is requested. All reports (analog and digital) contain the current value. The point is processed immediately with that value.

Optimizing the scanning

To optimize the scanning, maximize the size of the scan packets as follows:

- Disable Hiway Ownership if the HLPIU SOE reporting option is not being used.
- Choose a scanning period appropriate to the values being scanned. For example, you do not need to scan a temperature or level signal every 5 seconds if the response time of the process is several minutes.
- Specifying the minimum number of scan periods, because the server requires a separate scan for each scan period. For example, you might only need two scan periods for a particular controller, a short one for a few critical values, and a long one for all other values.

Defining TDC points

See the section Configuring the Enterprise Model in the *Configuration Guide* for information about Assets. For details on how to configure Assets before building the points, refer to the *Enterprise Model Builder User Guide*.

To define a point:

- 1 Click  to display the Add Items dialog.
- 2 Select Point as the item then select the Point Type:
 - **Accumulator Point.** To accumulate the changes to an analog value, for example, flow signals.
 - **Analog Point.** To monitor and control analog data and I/O signals, for example, PID loops.
 - **Composite Point.** To combine the data from multiple points.

- **Status Point.** To monitor and control status data and I/O signals, for example, motors and valves.
- 3 Click **OK** to add the point
- 4 See the *Quick Builder Reference* for a description of the property values and the “TDC point property settings” on page 16 further guidance.
- 5 After building several points, click  to download the point definitions to the server.



Note

The I/O slots associated with MCs, PIUs, DHPs and HIMs must be defined before the points using these slots can be successfully downloaded to the server, see the section “TDC I/O slot definitions” on page 9 for further information.

TDC point property settings

Tab	Property	Notes
Main	Point ID	Must be unique, can contain up to 40 alphanumeric characters, with at least one alpha character. Normally pre-fixed by a plant reference and named according to ISO or plant electrical standards, for example, 001FIC001.
	Area Code	Must be unique, can contain up to 40 alphanumeric characters, with at least one alpha character. Normally a plant reference, for example, CCU1.

Tab	Property	Notes
Main (continued)	PV Source Address	The controller name/address the server reads the value from. Either type the full address, or click  to specify the address with Address Builder. See “TDC parameter addresses” on page 21. If the address field is left blank the parameter is not displayed on the Group displays. If the point only has an output it is normal to specify the OP address for the PV parameter. For Status points with multiple inputs or outputs, the addresses must be consecutive and the only the lowest address is specified. The number input/output addresses is determined by the input and output state selections.
	PV Scan Period	See “Optimizing the scanning” on page 15 for guidance.
	Analog Point - Drift Deadband	The default is 0%, specify a drift deadband to reduce the system load, for example 0.1%.
	Status Point - Number of States	The number of binary states (2, 4 or 8) to display based on the number of inputs (1, 2 or 3). The input addresses must be consecutive.
	Status Point – State Descriptors	For single input TDC Data Hiway points, the state descriptors are normally 0 = Lower Box, 1 = Upper Box. For dual input TDC Data Hiway points, the state descriptors are normally 0 = Off, 1 = Upper Box, 2 = Lower Box, 3 = Fault.
Display	Group and Trend Displays	If the allocations are duplicated, the last point downloaded to the server takes the position in the group and trend displays. Caution: The display allocations for a point are deleted if the point is uploaded from the server. For more information, see “Configuration management” on page 2.

Tab	Property	Notes
Display (continued)	Point Detail Display	Special point detail display templates are available for TDC Extended Controller Analog points, see “Special TDC point and group displays” on page 36.
	Group Faceplate Template Display	A special group display templates is available for TDC Analog points, see “Special TDC point and group displays” on page 36.
Alarms	Control Failure Alarm	Sets the priority for all Control Failure Alarms, the default is Urgent. See the Control tab, Control Deadband and Control Timeout properties for a description of these alarms.
	External Change Alarms	When selected, an alarm is raised if the parameter changes without the change being initiated from the server. The default is not enabled. Do not enable these selections unless there is a good reason to do so because alarms can be generated by the normal control loop actions or when changes are made from another server or system.
	Status Point – Re-alarm on state transition	If selected, an alarm is raised whenever the PV changes to another alarm state, for example, if you make states 7 and 8 alarm states, an alarm is raised if the PV changes to state 7 and another alarm is raised if the state then changes to state 8. If this selection is disabled, only the first alarm is generated.
Control	Source Address	The controller name/address the server reads the value from. Either type the full address, or click  to specify the address with Address Builder. See “TDC parameter addresses” on page 21. If the address field is left blank the parameter is not displayed on the Group displays. For Status points with multiple inputs or outputs, the addresses must be consecutive and the only the lowest address is specified. The number input/output addresses is determined by the input and output state selections.

Tab	Property	Notes
Control (continued)	Destination Address	The controller name/address the server writes to. Either type the full address, or click  to specify the address with Address Builder. See “TDC parameter addresses” on page 21. If the address field is left blank the parameter is not displayed on the Group displays. For Status points with multiple inputs or outputs, the addresses must be consecutive and the only the lowest address is specified. The number input/output addresses is determined by the input and output state selections.
	Scan Periods	See “Optimizing the scanning” on page 15 for guidance.
	Reverse Output	This selection reverses the source and destination parameter values. Select if the device being controlled closes and opens on reverse signals, for example, to control a reverse acting valve.
	Control Confirmation	If selected, the operators are prompted to confirm the control action (Yes or No); the action is only carried out if they answer Yes.
	Analog Point - Output Limits	These limits are independent of the TDC box output limits and are applied in all modes (unlike the TDC output limits which are not active in manual mode). If the operator needs to be able to fully open or close the valve in manual mode the TDC output limits should be used and the point output limits should be disabled by setting the values to –10% and 110%.
	Analog Point - Setpoint Limits	These limits are independent of the TDC box setpoint limits. To use the TDC box setpoint limits, disable the points setpoint limits by setting the values outside of the configured range in engineering units.

Tab	Property	Notes
Control (continued)	Analog Point – Control Deadband	The percentage deviation from the entered value that represents good control, the default value is 1.000%. For MD changes, if the MD value does not achieve the MD setting within the Control Timeout period, a MD Control Failure alarm is generated. For OP changes, if the OP value does not achieve the OP setting +/- the Control Deadband within the Control Timeout period, an OP Control Failure alarm is generated. For SP changes, if the PV value does not achieve the SP setting +/- the Control Deadband within the Control Timeout period, a PV Control Failure alarm is generated. Note: This is not the same as a deviation alarm because PV deviations after the Timeout Period are not alarmed.
	Analog Point - Control Timeout	The maximum time (in seconds) allowed to achieve the target value or state before a Control Failure alarm is generated. The default is None (disabled), you must specify a value other than None to enable Control Failure alarms. The server checks the value every 10 seconds during the timeout period.
	Status Point – Pulse Width	Note: This setting is independent of the TDC digital output board settings.
	Status Point - Control Timeout	As for Analog points except: For OP changes, if the PV value does not achieve the OP state within the Control Timeout period, a PV Control Failure alarm is generated. Note: This is not the same as a TDC Command Disagree or Feedback alarm because control failures after the Timeout Period are not alarmed.
	Normal Mode	The default value is Auto. This setting can be used to indicate the normal control mode to the operators, manually controlled Status points should be set to Manual.

Tab	Property	Notes
Control (continued)	Status Point – Output State/ Target Input State Associations	See the <i>Quick Builder Guide</i> for a description of these settings. Note: A value of F means “not used”, you must specify F to disable the invalid control states.
Auxiliary		Used to read or write supplementary data to the controller.
History		Defines the history data collected for the point, there are limits to the number of point parameters that can be assigned to each history type, see the <i>Configuration Guide</i> for details.
User Defined		Used to add user defined parameters to the point, see the <i>Quick Builder Reference</i> for details.

TDC parameter addresses

The point parameter addresses consist of a controller name as well as an address location that depends on the box type. The addresses can be typed into the Source and Destination fields or specified using the Address Builder by clicking .



Note

Points cannot be built for the TDC Reserve Controller Directors (Basic, Multifunction and Extended).

Basic Controller (CB) addressing

The Basic Controller consists of 8 modulating slots. The address format is:

AS

where:

Part	Description
A	Is the address acronym listed in Table 1.
S	Is the slot number. See Table 1 for the valid range.

Example:

- $PV7$ (Process Variable for modulating slot 7)
- $K\ 7$ (Gain for modulating slot 7)
- $BI8$ (Bias for modulating slot 8)

Table 1 Basic Controller addresses

Address Parameter Description	Address Acronym (A)	Slot Range (S)
Analog point		
Process variable	PV (read only)	1 to 8
Setpoint	SP	1 to 8
Output	OP	1 to 8
Mode	MD	1 to 8
Gain	K (with a space)	1 to 8
Computer setpoint	CS	1 to 8
Computer output	CO	1 to 8
Digital filter	TD	1 to 8
Integral time	T1	1 to 8
Derivative time	T2	1 to 8
PV HI limit	PH*	1 to 8
PV LO limit	PL*	1 to 8
Deviation HI limit	DH*	1 to 8
Deviation LO limit	DL*	1 to 8
Remote variable	RV (read only)	1 to 8

Table 1 Basic Controller addresses (*continued*)

Address Parameter Description	Address Acronym (A)	Slot Range (S)
Bias	BI**	1 to 8
Ratio	RA**	1 to 8

* Only PH/PL or DH/DL is valid for any given subslot, depending on the slot configuration.

** The Ratio and Bias parameter should only be used with the PID Ratio and Bias algorithms.

Extended Controller (EC) addressing

The Extended Controller has 16 modulating slots, 16 digital outputs (8 SOA and 8 SOB), and 16 digital inputs. The address format is:

AS

where:

Part	Description
<i>A</i>	Is the address acronym listed in Table 2.
<i>S</i>	Is the slot number. See Table 2 for the valid range.

Examples:

- PV7 (Process Variable for modulating slot 7)
- K 7 (Gain for modulating slot 7)
- BI16 (Bias for modulating slot 16)
- SI12 (Status point on digital input slot 12)
- SOB6 (Status point on digital output B slot 6)

Table 2 Extended Controller addresses

Slot type	Address Parameter Description	Address Acronym (A)	Slot Range (S)
Status point			
Digital Input	Digital input	SI (read only)	1 to 16
Digital Output (A)	Digital output A	SOA	1 to 8
Digital Output (B)	Digital output B	SOB	1 to 8
Analog point			
Modulating	Process variable	PV (read only)	1 to 16
Modulating	Setpoint	SP	1 to 16

Table 2 Extended Controller addresses (*continued*)

Slot type	Address Parameter Description	Address Acronym (A)	Slot Range (S)
Modulating	Output	OP	1 to 16
Modulating	Mode	MD	1 to 16
Modulating	Gain	K (with a space)	1 to 16
Modulating	Integral time	TI	1 to 16
Modulating	Derivative time	T2	1 to 16
Modulating	Digital filter	TD	1 to 16
Modulating	PV HI limit	PH*	1 to 16
Modulating	PV LO limit	PL*	1 to 16
Modulating	Deviation HI limit	DH*	1 to 16
Modulating	Deviation LO limit	DL*	1 to 16
Modulating	Computer setpoint	CS	1 to 16
Modulating	Computer output	CO	1 to 16
Modulating	Z-input	Z (with a space, read only)	1 to 16
Modulating	Bias (algo 02, 03, and 04)	BI**	1 to 16
Modulating	Ratio (algo 02, 03, and 04)	RA**	1 to 16
Modulating	Manual PV (algo 20 and 30)	MV**	1 to 16
Modulating	Select input (algo21, 26, and 31)	IP**	1 to 16
Modulating	Segment number (algo 34)	SN**	1 to 16

* Only PH/PL or DH/DL is valid for any given subslot, depending on the slot configuration.

** These parameters should only be used with the specified algorithms.

Multifunction Controller addressing

The Multifunction Controller (MFC) and Advanced Multifunction Controller (AMC) appear functionally identical to the server. All addresses and parameters are the same.

The Multifunction Controller has modulating slots, sequence slots, logic block, flags, numerics, timers and Point Card File Assemblies (PCFAs) that can contain analog input, analog output, digital input, digital output and counter input modules.

MC modulating slot addresses

The address format is:

AS

where:

Part	Description
<i>A</i>	Is the address acronym listed in Table 3.
<i>S</i>	Is the slot number. See Table 3 for the valid range.

Examples:

- *PV7* (Process Variable for modulating slot 7)
- *K 7* (Gain for modulating slot 7)
- *BI16* (Bias for modulating slot 16)

Table 3 MC Modulating slot addresses

Address Parameter Description	Address Acronym (A)	Slot Range (S)
Analog point		
Process variable	PV (read only)	1 to 16
Setpoint	SP	1 to 16
Output	OP	1 to 16
Mode	MD	1 to 16
Gain	K (with a space)	1 to 16
Integral time	T1	1 to 16
Derivative time	T2	1 to 16
Digital filter	TD	1 to 16
PV HI limit	PH*	1 to 16
PV LO limit	PL*	1 to 16

Table 3 MC Modulating slot addresses (*continued*)

Address Parameter Description	Address Acronym (A)	Slot Range (S)
Deviation HI limit	DH [*]	1 to 16
Deviation LO limit	DL [*]	1 to 16
Computer setpoint	CS	1 to 16
Computer output	CO	1 to 16
Bias	BI ^{**}	1 to 16
Ratio	RA ^{**}	1 to 16

* Only PH/PL or DH/DL is valid for any given subslot, depending on the slot configuration.

** The Ratio and Bias parameters should only be used with the PID Ratio and Bias algorithms.

MC Point Card File Assembly (PCFA) slot addresses

The address format is:

ASS

where:

Part	Description
<i>A</i>	Is the address acronym, listed in Table 4.
<i>S</i>	Is the slot number. See Table 4 for the valid range.
<i>s</i>	Is the subslot number. See Table 4 for the valid range.



Note

PCFA slots 1 to 16 are addressed as slots 17 to 32 (for example, PCFA 7 is server slot 32). This is to avoid confusion with modulating slots 1 to 16.

Examples:

- do2512 (digital output on slot 25, subslot 12)
- pv2301 (Analog input on slot 23, subslot 1)

Table 4 MC PCFA slot addresses

Slot Type	Address Parameter Description	Address Acronym (A)	Slot Range (S)	Subslot Range (s)
Status point				
DI	Digital input	DI	17 to 32	1 to 16

Table 4 MC PCFA slot addresses (*continued*)

Slot Type	Address Parameter Description	Address Acronym (A)	Slot Range (S)	Subslot Range (s)
DO	Digital output	DO	17 to 32	1 to 8
	Mode	MD	17 to 32	1 to 8
Analog point				
AI	Process variable	PV	23 to 24	1 to 8
	Setpoint	SP	23 to 24	1 to 8
	PV HI limit	PH*	23 to 24	1 to 8
	PV LO limit	PL*	23 to 24	1 to 8
	Deviation HI limit	DH*	23 to 24	1 to 8
	Deviation LO limit	DL*	23 to 24	1 to 8
CI	Process variable	PV	17 to 32	1 to 4
	Setpoint	SP	17 to 32	1 to 4
AO	Output	OP	17 to 32	1 to 4
	Mode	MD	17 to 32	1 to 4
Accumulator point				
CI	Process variable	PV	17 to 32	1 to 4

* Only PH/PL or DH/DL is valid for any given subslot, depending on the slot configuration.

Sequence slot addresses

The points that address the sequence slots are also called CL/MC Process Module Points. See “Process module points” on page 3. The address format is:

SEQnn variable

where:

Part	Description
<i>nn</i>	Is the two digit sequence slot number.
<i>variable</i>	Is the sequence variable listed in Table 5.

Example:

SEQ12 STATE (State parameter for sequence slot 12).

Table 5 MC Sequence slot addresses

Sequence Variable	Description
Status Point	
REQ	request (the only writable variable)
RSTR	restart enabled
HOLD	hold enabled
SHDN	shutdown enabled
EMSD	emergency shutdown enabled
MODE	sequence mode
STATE	sequence state
UNIT	unit state
ALARM	phase alarm
CFM	message confirm
Analog Point	
FAIL	fail code
MEM	memory block number
NUMBER	sequence number
PHASE	phase number
NSTEP	normal step number
NSTMT	normal statement number
SSTEP	subroutine step number
SSTMT	subroutine statement number
ASTEP	abnormal step number
ASTMT	abnormal statement number
TIMER	phase alarm timer

Logic, numeric, flag and timer addresses

The address format is:

*A**n*

where:

Part	Description
<i>A</i>	Is the address acronym listed in Table 6.
<i>n</i>	Is the Parameter number. See Table 6 for the valid range.

Examples:

- LG116 (Logic block 116)
- NV27 (Numeric 27)

**Note**

See the *SOPL User's Manual* for information on configuring the logic blocks.

Table 6 MC box variable addresses

Address Parameter Description	Address Acronym (A)	Parameter Number (n)
Status point		
Logic block variable	LG	1 to 128
Flag variable	FG	1 to 256
Analog point		
Numeric variable	NV	1 to 88
Timer variable	TV	1 to 32
Timer setpoint	TS	1 to 32

Process Interface Unit (PIU) addressing

The HLPIU consists of up to 32 slots containing analog input, analog output, digital input, digital output, or counter input cards.

The LLPIUs and LEPIUs consists of up to 32 slots of analog inputs.



Note

Subslot 8 for slots 16 and 32 are not available as AI.

The address format is:

ASS

where:

Part	Description
<i>A</i>	Is the address acronym listed in Table 7 and Table 8.
<i>S</i>	Is the slot number. See Table 7 and Table 8.
<i>s</i>	Is the subslot number. See Table 7 and Table 8.

Examples:

- PV2307 (Process Variable for slot 23 subslot 7)
- DI0413 (Digital Input slot 4 subslot 7)

HLPIU addresses

Table 7 HLPIU addresses

Slot type	Address Parameter Description	Address Acronym	Slot Range	Subslot Range
Status point				
Change Detect	Digital input	DI	1 to 32	1 to 16
Notify Status	Digital input	DI	1 to 32	1 to 16
Sequence of Events	Digital input	DI	1 to 32	1 to 16
Pulsed Output	Digital output	DO	1 to 32	1 to 8
Latched Output	Digital output	DO	1 to 32	1 to 8
	Mode	MD	1 to 32	1 to 8
Analog point				
AI	Process variable	PV	1 to 32	1 to 8

Table 7 HLIU addresses (*continued*)

Slot type	Address Parameter Description	Address Acronym	Slot Range	Subslot Range
	Setpoint	SP	1 to 32	1 to 8
	PV HI Limit	PH [*]	1 to 32	1 to 8
	PV LO Limit	PL [*]	1 to 32	1 to 8
	Deviation HI Limit	DH [*]	1 to 32	1 to 8
	Deviation LO Limit	DL [*]	1 to 32	1 to 8
CI (16-bit)	Process variable	PV	1 to 32	1 to 8
	Setpoint	SP	1 to 32	1 to 8
CI (32-bit)	Not supported			
	Setpoint	SP	1 to 32	1 to 4
AO	Output	OP	1 to 32	1 to 4
	Mode	MD	1 to 32	1 to 4
Accumulator Point				
AI	Process variable	PV	1 to 32	1 to 8
CI (16-bit)	Process variable	PV	1 to 32	1 to 8
	Setpoint	SP	1 to 32	1 to 8
CI (32-bit)	Not supported			

* Only PH/PL or DH/DL is valid for any given subslot, depending on the slot configuration.

LLPIU and LEPIU addresses

Table 8 LLPIU and LEPIU address parameters and slot range

Slot type	Address Parameter Description	Address Acronym (A)	Slot Range (S)	Subslot Range (s)
Analog point				
AI	Process variable	PV	1 to 32	1 to 8
	PV HI limit	PH [*]	1 to 32	1 to 8
	PV LO limit	PL [*]	1 to 32	1 to 8
	Deviation HI limit	DH [*]	1 to 32	1 to 8
	Deviation LO limit	DL [*]	1 to 32	1 to 8
	Setpoint	SP	1 to 32	1 to 8

Table 8 LLPIU and LEPIU address parameters and slot range (*continued*)

Slot type	Address Parameter Description	Address Acronym (A)	Slot Range (S)	Subslot Range (s)
Accumulator point				
AI	Process variable	PV	1 to 32	1 to 8

* Only PH/PL or DH/DL is valid for any given subslot, depending on the slot configuration.

Data Hiway Port (DHP) and LCS 620 HIM addressing

The Data Hiway Port has up to 32 slots. Slots 16 and 32 are reserved and cannot be addressed by the server. Slots 1 to 15 and 17 to 31 can be analog input, analog output, digital input, digital output, or counter inputs.

A 620LCS PLC connected to the TDC Data Hiway by way of a Hiway Interface Module (HIM) will appear as up to four Data Hiway Ports (DHPs). The address configuration for the 620 LCS is the same as for the DHP.

The address format is:

ASS

where:

Part	Description
<i>A</i>	Is the address acronym listed in Table 9.
<i>S</i>	Is the slot number. See Table 9 for the valid range.
<i>s</i>	Is the subslot number. See Table 9 for the valid range.

Examples:

- PV237 (Process Variable for slot 23 subslot 7)
- DI043 (Digital input on slot 4, subslot 3)

Table 9 DHP/HIM addresses

Slot type	Address Parameter Description	Address Acronym	Slot Range	Subslot Range
	Status Point			
Notify Status	Digital input	DI	1 to 15, 17 to 31	1 to 16
Latched Output	Digital output	DO	1 to 15, 17 to 31	1 to 8
	Mode	MD	1 to 15, 17 to 31	1 to 8
Analog Point				

Table 9 DHP/HIM addresses (*continued*)

Slot type	Address Parameter Description	Address Acronym	Slot Range	Subslot Range
AI	Process variable	PV	1 to 15, 17 to 31	1 to 6
	Setpoint	SP	1 to 15, 17 to 31	1 to 6
	PV HI limit	PH*	1 to 15, 17 to 31	1 to 6
	PV LO limit	PL*	1 to 15, 17 to 31	1 to 6
	Deviation HI limit	DH*	1 to 15, 17 to 31	1 to 6
	Deviation LO limit	DL*	1 to 15, 17 to 31	1 to 6
CI	Process variable	PV	1 to 15, 17 to 31	1 to 8
AO	Output	OP	1 to 15, 17 to 31	1 to 4
	Mode	MD	1 to 15, 17 to 31	1 to 4
Accumulator Point				
AI	Process variable	PV	1 to 15, 17 to 31	1 to 6
CI	Process variable	PV	1 to 15, 17 to 31	1 to 8

* Only PH/PL or DH/DL is valid for any given subslot, depending on the slot configuration.

Hardware generic addressing

Hardware generic addressing can be used to access data not supported by the standard address acronyms. This form of addressing must be used with caution because it is possible to corrupt the box database by reading or writing to the wrong addresses.

See the TDC manuals listed in “Referenced documentation” on page 2 to identify the required addresses.

Hardware generic addresses are specified in either octal, hexadecimal, or decimal syntax. The TDC manuals list the addresses in octal.



Note

If using the B12S format, the server displays the value in engineering units, not in percent. The BOS/EOS will display this value in percent.

The address format is:

H:O'xyy B ww format

Part	Description
<i>xx</i>	Slot number minus 1 in octal (for example, slot 10 = 0'11)
<i>yy</i>	Address in octal as defined in the TDC manuals
<i>B</i>	Starting bit number
<i>ww</i>	Width from the starting bit
<i>format</i>	Data format (as defined in the TDC manuals): • B0 • B4 • B7 • B12 • B15 • B12E • B12S

Example:

H:0'0716 0 16 B4 (For slot 8, MFC modulating slot auxiliary constant at NN16 with format B4).

TDC engineering unit conversions

For CB, EC, MC and PIU points with TDC Engineering Unit Conversion (EUC) configured, the server automatically detects the existence of the EUC and performs linearization using the appropriate piecewise, fourth-order polynomial algorithms.



Note

For the linearization to take place, the server point must not have any other PV algorithm attached.

The linearization algorithm is automatically selected and attached to the server point. The algo numbers reserved for linearization are 101 to 115 and 121 to 135 corresponding to EUCs 1 to 15 in degrees Fahrenheit and Celsius respectively.

Thermocouple Linearization EU Conversions

PV algo	TDC EUC	Description	Valid Range (Deg C)	Valid Range (Deg F)
101/121	1	type J thermocouple	-75 to 1095	-103 to 2003
102/122	2	type K thermocouple	-75 to 1374	-103 to 2505
103/123	3	type T thermocouple	-186 to 400	-303 to 752
104/124	4	type S thermocouple	-19 to 1762	-2 to 3203
105/125	5	Square root	N/A	N/A
106/126	6	platinum RTD Burns	-186 to 650	-303 to 1202
107/127	7	RH radiamatic	592 to 1762	1097 to 3203
108/128	10	type B thermocouple	798 to 1704	1468 to 3099
109/129	11	type E thermocouple	-202 to 1002	-331 to 1835
110/130	12	type R thermocouple	-50 to 1760	-58 to 3200
111/131	13	type R thermocouple (JIS)	0 to 1770	32 to 3218
112/132	14	platinum RTD DIN	-212 to 649	-349 to 1200
113/133	15	platinum RTD JIS	-200 to 630	-382 to 1166

Special TDC point and group displays

The following TDC displays are installed as part of the Station software:

Display Filename	Description
sysDtlAnaGroup_TDC.dsp	This Analog Group display includes a Normal Mode indication for Analog points. To use this display, click the Display tab on the Analog point and change the Group Faceplate Template Display entry to sysDtlAnaGroup_TDC.
sysDtlAna_TDC_ECPID.dsp	This Analog Point Detail display can be used with Extended Controller modulating slots configured to use one of the PID algorithms without Ratio and Bias, that is, 01 and 05 to 13. To use this display, click the Display tab on the primary Analog point and change the Point Detail Display entry to sysDtlAna_TDC_ECPID.
sysDtlAna_TDC_ECPIDRB.dsp	This Analog Point Detail display can be used with Extended Controller modulating slots configured to use one of the PID Ratio and Bias algorithms, that is, 02 to 04. The point must be configured as described in “Point configuration for TDC EC PIDRB display” on page 36.
sysDtlAna_TDC_EC.dsp	This Analog Point Detail display can be used with Extended Controller modulating slots configured to use one of the auxiliary algorithms, that is, 00 and 14 to 37. These algorithms support independent X and Y input ranges, which makes it necessary to build two analog points as described in “Point configuration for TDC EC display” on page 37.
sysDtlAna_TDC_Template.dsp	This display is the master template used to create the above displays, it has been included to allow other displays to be created if required.

Point configuration for TDC EC PIDRB display

The following analog point configuration should be used with Extended Controller modulating slots configured to use one of the PID Ratio and Bias algorithms, that is, 02 to 04 when using the sysDtlAna_TDC_ECPIDRB display:

Tab	Property	Entry	Notes
Display	Point Detail Display	sysDtlAna_TDC_ECPIDRB	
	Group Faceplate Template Display	sysDtlAnaGroup_TDC	Optional

Tab	Property	Entry	Notes
Auxiliary	A1 Parameter Name	Ratio	This name is used by the display
	A1 Source & Destination Address	See “Extended Controller (EC) addressing” on page 23	
	A2 Parameter Name	Bias	This name is used by the display
	A2 Source & Destination Address	See “Extended Controller (EC) addressing” on page 23	

Point configuration for TDC EC display

The following analog point configuration should be used with Extended Controller modulating slots configured to use one of the auxiliary algorithms, that is, 00 and 14 to 37. These algorithms support independent X and Y input ranges, which makes it necessary to build two analog points for the slot.



Note

A points SP value is not linearized according to the regulatory slot Y input configuration setting. When linearization is configured on the X input, a PV algorithm is automatically added to the associated point and linearization is applied to the points PV and SP values. This functionality is not supported for the Y input linearization setting. If linearization is configured on the X input the same setting should be used on the Y input. Points with a SP parameter but no PV parameter cannot be linearized.

Primary PV/X Input Point:

Tab	Property	Entry	Notes
Main	PV Source Address	See “Extended Controller (EC) addressing” on page 23	PV/X input source address
	Engineering Units	As required	PV/X input engineering units
Display	Point Detail Display	sysDtlAna_TDC_EC	
	Group Faceplate Template Display	sysDtlAnaGroup_TDC	Optional
Control	SP Source & Destination Address	Leave blank	

Tab	Property	Entry	Notes
	OP Source & Destination Address	See “Extended Controller (EC) addressing” on page 23	
	MD Source & Destination Address	See “Extended Controller (EC) addressing” on page 23	
Auxiliary	A1 Parameter Name	YinXeu	This name is used by the display
	A1 Source & Destination Address	SP parameter address, see “Extended Controller (EC) addressing” on page 23	This value is used by the Y input bar
User Defined	Parameter Name	Yinput	This name is used by the display
	Link Type	Parameter Reference	
	Point Name	Enter secondary point name	Primary point name with suffix Y
	Parameter Name	SP	

Secondary SP/Y Input Point:

Tab	Property	Entry	Notes
Main	PV Source Address	Leave blank	PV/X input source address
	Engineering Units	As required	SP/Y input engineering units
Display	Point Detail Display	default	
	Group Faceplate Template Display	sysDtlAnaGroup_TDC	Optional
Control	SP Source & Destination Address	See “Extended Controller (EC) addressing” on page 23	
	OP Source & Destination Address	Leave blank	
	MD Source & Destination Address	Leave blank	

Additional tasks

To complete the TDC Box configuration, see the *TDC BOS/EOS Configuration Form Instructions* and the controller algorithms reference manuals.

For information on configuring the Multifunction Controller Logic Blocks, see the *TDC SOPL User's Manual*.

If CL/MC Sequence Programs are being used, see “Control Language reference” on page 1.

See the *Configuration Guide* for all other tasks, for example, displays and reports.

System operations

5

This chapter contains the information required to monitor and control the TDC Data Hiway boxes from the Experion system. For general information refer to the *Operator's Guide* listed in the section “Referenced documentation” on page 2.

For information about:	Go to:
Monitoring the System Status	page 2
TDC Data Hiway Commands	page 4
TDC Error Codes	page 7
TDC Sequence Programs	page 53

Monitoring the system status



Note

The indicators on the Channel and Controller summary displays represent the communication status; for the TDC box status and box error descriptions use the Controller Configuration displays.

The following table provides a summary of the main failure modes, their symptoms, causes and consequences:

Symptom	Possible Causes	Consequences	Actions
Station disconnect	Redundant Server fail over	Temporary loss of process view and control.	The Server fail over time is typically 30 seconds, the Stations should reconnect automatically.
	Server maintenance or network failure	Loss of process view and control.	Investigate and correct the problem. Perform Station reconnects.
Channel COMMS alarm	Communication failure between the Server and the Data Hiway interface device	If the Channel is redundant, the communications will fail over to the redundant Channel without loss of process view and control.	Investigate and correct the problem.
		If the Channel is not redundant, loss of process view and control.	May require manual Server fail over depending on the system architecture.
H2 Controller COMMS alarm	TDC device powered down	Partial loss of process view and control	Investigate and correct the problem.
	Data Hiway cable fault	Partial loss of process view and control.	Switch to the backup Data Hiway cable using the SWT <i>nn</i> command or HTD switch panel.

Symptom	Possible Causes	Consequences	Actions
A80 Controller COMMS alarm	Recovery from power loss	Partial loss of process view and control.	Use CLR <i>nn</i> command for PIU, DHP and HIM devices. Use STR <i>nn</i> command for CB, EC, MC and RCD devices. The box database may need to be download (restored), this requires the Supervisor level login.
RSET Controller COMMS alarm	TDC device in Reset State	Partial loss of process view and control.	Use CLR <i>nn</i> command for PIU, DHP and HIM devices. Use STR <i>nn</i> command for CB, EC, MC and RCD devices.
IDLE Controller COMMS alarm	MC in IDLE State	Partial loss of process view and control.	Use PRC <i>nn</i> command to change to processing state.
Other Controller COMMS alarms	Controller error	Partial loss of process view and control.	Call up the Controller Detail displays for box status and error description. Refer to the TDC manuals for possible cause and suggested corrective actions.
CONFIG MISMTCH alarm	The Server configuration does not match the TDC box configuration.	The associated point detail display values may be incorrect.	Investigate and correct the configuration.

TDC Data Hiway commands

The Data Hiway commands are used to switch to the backup Data Hiway cable and to control the state of the Data Hiway boxes.



Note

The Data Hiway commands use the Controller Number as displayed on the Controller Status display, this number can be different to the Data Hiway box address used by the BOS and EOS stations.

The Data Hiway commands are entered in the Station Command Zone. The message “Control Executed Successfully” is displayed when the command is successfully delivered to the Data Hiway box. After entering a command, the status of the Data Hiway box should be checked on the Controller Configuration displays.



Note

The BOS/EOS INIT command is not supported; use the CLR *nn* command for PIU, DHP and HIM devices and the STR *nn* command for CB, EC, MC and RCD devices.

Data Hiway switch command



Note

The HTD can only switch cables once, the switch back must be performed via HTD panel.

Command	Description
SWT <i>nn</i>	Request DHB or HTD to switch to the backup Data Hiway cable.

Basic Controller (CB) and Extended Controller (EC) commands

Command	Description
STR <i>nn</i>	Restart controller <i>nn</i> and initialize the box address locations in the controller.

Multifunction Controller (MC) commands

Command	Description
STR <i>nn</i>	Restart controller <i>nn</i> and initialize the box address locations in the controller.
PRC <i>nn</i>	Change the state of controller <i>nn</i> to PROCESSING.
IDL <i>nn</i>	Change the state of controller <i>nn</i> to IDLE.

Reserve Controller Director (RCD) commands

Command	Description
STR <i>dd</i>	Restart Reserve Controller Director <i>dd</i> and initialize the box address locations in the controller.
STR <i>dd</i> 00	Restart the Reserve Controller associated with Reserve Controller Director <i>dd</i> .
STR <i>dd nn</i>	Restart Primary Controller <i>nn</i> through Reserve Controller Director <i>dd</i> .
TST <i>nn</i>	Test backup (switch) from Primary Controller <i>nn</i> to its associated Reserve Controller.
SAV <i>nn</i>	Back up (switch) from Primary Controller <i>nn</i> to its associated Reserve Controller.
ENB <i>nn</i>	Enable back up of Primary Controller <i>nn</i> .
DIS <i>nn</i>	Disable back up of Primary Controller <i>nn</i> .
SWB <i>dd</i>	Switch back from Reserve to the Primary Controller through the Reserve Controller Director <i>dd</i> .

PIU, DHP and HIM commands

Command	Description
CLR <i>nn</i>	Enable processing for controller <i>nn</i> and initialize the box address locations in the controller.
STR <i>nn</i>	This command clears the controller memory and restarts controller <i>nn</i> . The box database will normally need to be Downloaded (restored) after using this command.

CLR nn command

If the server is a Hiway Owner, set the:

- Change rate
- Alarm device
- Significant change option

Clear alarm report, general report and charge report.

Enable processing for controller nn.

Wait for watchdog timer to start:

- HLPIU = 0.5 seconds
- LLPIU = 11.5 seconds
- LEPIU = 22.5 seconds

Initialize the box address for controller nn.

STR nn command

Clears the box database.

Restart controller nn.

If the server is Hiway Owner, set the:

- Change rate
- Alarm device
- Significant change option

Wait for watchdog timer to start:

- HLPIU = 0.5 seconds
- LLPIU = 11.5 seconds
- LEPIU = 22.5 seconds

Set box address.

TDC error codes

The following error codes may be displayed in the alarm messages and on the Controller Status display; the TDC box status and error descriptions are displayed on the Controller Configuration displays.

For information about	Go to
Hiway addressing/communication error codes	page 8
HLPIU (3xx), LLPIU (4xx) and LEPIU (5xx) error codes	page 13
DHP or 620 HIM error codes	page 17
Basic controller primary (1xx) and reserve (6xx) error codes	page 21
Extended controller primary (7xx) and reserve (8xx) error codes	page 23
Multifunction controller (MFC and AMC) error codes	page 27
UAC primary controller codes	page 36
UAC reserve controller Director CB (6xx) and EC (8xx) error codes	page 38
UAC reserve controller Director MC error codes	page 41
UAC reserve controller CB (6xx) and EC (8xx) error codes reported by the RCD	page 45
UAC reserve controller MC error codes reported by the RCD	page 49

Hiway addressing/communication error codes

Error	A80
Description	Address error on devices internal address word(s).
Cause	
Solution	For PIU/DHP/HIM use CLR command to initialize hiway address. For CB, EC, MC use STR command. If the problem persists check the Hiway I/F cards or RAM board.

Error	A81
Description	Two devices responding to the same address.
Cause	Proper device is pre-empted by other responding box or device communications inhibited by other responding device.
Solution	Take device off line before restoring Hiway. This write-enables to the pre-empted device.

Error	Axx
Description	Device number assumed by the faulty device.
Cause	Where xx (1 to 63) indicates the address of another device (which is also responding to the address where this code appears) that is, Hiway status display-25 1M-A24 (box 25 responds to the same address as box 24).
Solution	Service box at indicated address (Axx) by replacing Hiway I/F cards.

Error	H1
Description	No Hiway grant from the HTD.
Cause	No Hiway grant given to DHB or DHEB.
Solution	

Error	H2
Description	No Hiway response from the device.
Cause	
Solution	Check file power supply, Data Hiway cables, address links and Data Hiway I/F cards.

Error	H4
Description	BCH code error.
Cause	BCH check failure during communication.
Solution	Check file power supply or replace Hiway I/F cards.

Error	H5
Description	Busy response.
Cause	Device busy or PIU/ DHP/HIM WDT might be timed out.
Solution	Service PIU/DHP/HIM.

Error	H6
Description	No bus acknowledgment at Hiway I/F.
Cause	
Solution	

Error	H7
Description	Echo error.
Cause	Echo checking error during communication.
Solution	

Error	H8
Description	No response from Hiway I/F.
Cause	
Solution	

Error	H9
Description	Improper command word.
Cause	
Solution	

Error	H10
Description	Illegal response.
Cause	
Solution	

Error	H11
Description	Hiway driver transmit buffer overflow.
Cause	
Solution	

Error	H12
Description	BCH error on addressing diagnostic.
Cause	
Solution	

Error	H13
Description	Illegal response on addressing diagnostic.
Cause	
Solution	

Error	H14
Description	No response on addressing diagnostic.
Cause	
Solution	

Error	H15
Description	Device address error.
Cause	Device address compare check (primary) did not match box address and complement as expected.
Solution	

Error	H16
Description	Allowable time on Hiway exceeded.
Cause	
Solution	

Error	H17
Description	Secondary box failure.
Cause	Secondary box failure due to an H15 condition; box address obtained on H15 error points to this box.
Solution	

Error	H18
Description	Device internal memory address error.
Cause	
Solution	

Error	H19
Description	Stray interrupt failed Hiway.
Cause	
Solution	

Error	H31
Description	Other error.
Cause	Error other than above.
Solution	

HLPIU (3xx), LLPIU (4xx) and LEPIU (5xx) error codes

Error	x00
State	Reset
Description	Watch dog timer (WDT) is running but PIU is not processing I/O data.
Solution	This condition can appear temporarily when the PIU configuration data is changed.

Error	x00
State	WDT fail.
Description	Watch dog timer (WDT) has expired.
Solution	

Error	x51
State	PIU Fail.
Description	A/D converter failure.
Solution	Service A/D or MUX I/O card in the master file.

Error	x52
State	PIU Fail.
Description	PIU processing overload, use CLR command to reset the PIU and re-enable processing.
Solution	If x52 condition persists, put some of the points on a slower scan. See PIU LEDs (special LED bank on CCFA RAM Boards 4th LED from top). If LED is ON x52 error is still current.

Error	x53
State	PIU Fail.
Description	A/D zero offset exceeds limits.
Solution	Service A/D card in master file.

Error	x54
State	PIU Fail.
Description	Multiple relays pulled.
Solution	

Error	x55
State	PIU Fail.
Description	Point card zero failure.
Solution	

Error	x56
State	PIU Fail.
Description	No zero offset error.
Solution	

Error	x96
State	Undefined.
Description	Undefined error.
Solution	Probable cause is failure of the Hiway I/F card; Continuous Notification Writes from the box and no response to Callup.

Error	x98
State	Alarm Fail.
Description	PIU reporting failure.
Solution	

Error	x99
State	Alarm Fail.
Description	Alarm reporting device changed.
Solution	

Error	301 to 332
State	PIU Fail.
Description	HLPIU I/O card failure 01 to 32, the error code indicates the first failed card. The display shows a list of failed cards.
Solution	Service I/O cards.

Error	350
State	PIU Fail.
Description	Point Card File power loss.
Solution	

Error	390
State	PIU Fail.
Description	Sequence of events (SOE) drift.
Solution	

Error	401 to 432
State	PIU Fail.
Description	LLPIU MUX relay failure 01 to 32, the error code indicates the first MUX Box failure. The display shows a list of failed boxes.
Solution	Service MUX box.

Error	450
State	PIU Fail.
Description	LLPIU MUX relay stuck, RTD power failure or reference board missing or bad offsets.
Solution	Service the sub-multiplexer relay board.

Error	461
State	PIU Fail.
Description	LLPIU cold junction reference failure.
Solution	Service CJR-RTD on the 2nd Point of the first strip of the first PCFA.

Error	501 to 516
State	PIU Fail.
Description	LEPIU remote MUX Box failure 01 to 16, the error code indicates the first MUX Box failure. The display shows a list of failed boxes.
Solution	Service remote MUX box.

Error	550
State	Reset.
Description	MUX board offset or stuck relay: 1. Inability to establish offsets in a MUX board 2. Failure to recover from a stuck relay indication 3. Offsets out of limits while enable processing
Solution	Service A/D or MUX I/O card in the master file.

Error	561 to 576
State	PIU Fail.
Description	LEPIU cold junction resistor (CJR) failure in LEPIU MUX box 1 to 16. The error code lists the worst most recent failure.
Solution	Service CJR.

DHP or 620 HIM error codes

Error	1100
State	Reset.
Description	Watch dog timer (WDT) is running but DHP is not processing I/O data.
Solution	This condition can appear temporarily when the DHP configuration data is changed.

Error	1101
State	WDT Fail.
Description	Watch dog timer (WDT) has expired.
Solution	

Error	1102
State	DHP Fail.
Description	Link card CPU test failure (DHP only).
Solution	

Error	1103
State	DHP Fail.
Description	Link card ROM test failure (DHP only).
Solution	

Error	1104
State	DHP Fail.
Description	Link card RAM test failure (DHP only).
Solution	

Error	1105
State	DHP Fail.
Description	Link card Initialization failure (DHP only).
Solution	

Error	1106 to 110F
State	DHP Fail.
Description	Unknown failure (DHP only).
Solution	

Error	1191
State	Bat Fail.
Description	Memory battery failure, device status remains OK. Also displays slot 16 failure.
Solution	

Error	1192
State	Overload.
Description	Processing overload, actual scan time exceeds configured scan time.
Solution	

Error	1193
State	Saved.
Description	Primary 620 LCS HIM failure.
Solution	

Error	1194
State	RC Fail.
Description	Backup 620 LCS HIM failure.
Solution	

Error	1195
State	Soft Fail.
Description	620 HIM firmware not at correct revision.
Solution	

Error	1196
State	Undefine.
Description	Undefined error.
Solution	Probable cause is failure of the Hiway I/F; Continuous Notification Writes from the box and no response to Callup.

Error	1198
State	Alarm Fail.
Description	Reporting failure.
Solution	

Error	1199
State	Alarm Fail.
Description	Alarm reporting device changed.
Solution	

Error	11YX
State	
Description	For DHP: Y = device # 1 to 8 X = 1 to 4, where: 1 = Device link failure 2 = Device box or communications error 3 = Device configuration error 4 = Unknown device error For 620 HIM: Y = Device # 1 X = 1 or 3, where: 1 = Computer link failure 3 = Configuration error
Solution	

Basic controller primary (1xx) and reserve (6xx) error codes

Error	n00
State	Reset.
Description	Watch dog timer (WDT) expired. Try to restart (STR command).
Solution	Where possible, put the affected loops in Manual mode. See TDC maintenance manual.

Error	n01
State	Reset.
Description	A/D 0% calibration point exceeds +/- 1% limit.
Solution	Where possible, put the affected loops in Manual mode. See TDC maintenance manual.

Error	n02
State	Reset.
Description	A/D 100% calibration point exceeds +/- 1% limit.
Solution	Where possible, put the affected loops in Manual mode. See TDC maintenance manual.

Error	n03
State	Reset.
Description	Output card 1 failure control slots 1 to 4.
Solution	Where possible, put the affected loops in Manual mode. See TDC maintenance manual.

Error	n04
State	Reset.
Description	Output card 2 failure control slots 5 to 8.
Solution	Where possible, put the affected loops in Manual mode. See TDC maintenance manual.

Error	n05
State	Reset.
Description	Sample time clock (STC) failure.
Solution	Where possible, put the affected loops in Manual mode. See TDC maintenance manual.

Error	n06
State	Reset.
Description	A/D conversion time too slow.
Solution	Where possible, put the affected loops in Manual mode. See TDC maintenance manual.

Error	n07
State	Reset.
Description	CPU instruction check failure.
Solution	Where possible, put the affected loops in Manual mode. See TDC maintenance manual.

Error	n08
State	
Description	Sample time clock (STC) Integrity Check failure.
Solution	Where possible, put the affected loops in Manual mode. See TDC maintenance manual.

Error	n09
State	Reset.
Description	CMOS backup memory failure.
Solution	Where possible, put the affected loops in Manual mode. See TDC maintenance manual.

Extended controller primary (7xx) and reserve (8xx) error codes

Error	n00
State	Reset.
Description	Watch dog timer (WDT) expired. Try to restart (STR command).
Solution	Where possible, put the affected loops in Manual mode. See TDC maintenance manual.

Error	n01
State	Soft Failure.
Description	A/D drift.
Solution	Where possible, put the affected loops in Manual mode. See TDC maintenance manual.

Error	n02
State	
Description	Undefined error.
Solution	Where possible, put the affected loops in Manual mode. See TDC maintenance manual.

Error	n03
State	Soft Failure.
Description	Output card 1 failure control slots 1 to 4.
Solution	Where possible, put the affected loops in Manual mode. See TDC maintenance manual.

Error	n04
State	Soft Failure.
Description	Output card 2 failure control slots 5 to 8.
Solution	Where possible, put the affected loops in Manual mode. See TDC maintenance manual.

Error	n05
State	Reset.
Description	Sample time clock (STC) failure.
Solution	Where possible, put the affected loops in Manual mode. See TDC maintenance manual.

Error	n06
State	Reset.
Description	A/D Failure.
Solution	Where possible, put the affected loops in Manual mode. See TDC maintenance manual.

Error	n07
State	Reset.
Description	CPU instruction check failure.
Solution	Where possible, put the affected loops in Manual mode. See TDC maintenance manual.

Error	n08
State	
Description	Sample time clock (STC) Integrity Check failure.
Solution	Where possible, put the affected loops in Manual mode. See TDC maintenance manual.

Error	n09
State	Reset.
Description	RAM hard failure.
Solution	Where possible, put the affected loops in Manual mode. See TDC maintenance manual.

Error	n10
State	Reset.
Description	ROM checksum failure.
Solution	Where possible, put the affected loops in Manual mode. See TDC maintenance manual.

Error	n11
State	
Description	DEP interface failure.
Solution	Where possible, put the affected loops in Manual mode. See TDC maintenance manual.

Error	n12
State	
Description	Backup memory failure.
Solution	Where possible, put the affected loops in Manual mode. See TDC maintenance manual.

Error	n13
State	Soft Failure.
Description	RAM soft failure.
Solution	Where possible, put the affected loops in Manual mode. See TDC maintenance manual.

Error	n23 to 38
State	
Description	Transmitter failures, 01-16.
Solution	Where possible, put the affected loops in Manual mode. See TDC maintenance manual.

Error	n90
State	
Description	Alarm circular list full.
Solution	

Error	n97
State	
Description	Alarm ACK device changed.
Solution	

Multifunction controller (MFC and AMC) error codes

Error	900
State	Reset.
Description	Watch dog timer (WDT) expired. Try to restart (STR command).
Solution	Where possible, put the affected loops in Manual mode. See TDC maintenance manual.

Error	902
State	Reset.
Description	SBC overload.
Solution	Where possible, put the affected loops in Manual mode. See TDC maintenance manual.

Error	905
State	Reset.
Description	Sample time clock (STC) failure.
Solution	Where possible, put the affected loops in Manual mode. See TDC maintenance manual.

Error	907
State	Reset.
Description	CPU instruction check failure.
Solution	Where possible, put the affected loops in Manual mode. See TDC maintenance manual.

Error	908
State	
Description	Sample time clock (STC) Integrity failure.
Solution	Where possible, put the affected loops in Manual mode. See TDC maintenance manual.

Error	909
State	Reset.
Description	Standard RAM card read/write failure.
Solution	Where possible, put the affected loops in Manual mode. See TDC maintenance manual.

Error	910
State	Reset.
Description	ROM card checksum failure.
Solution	Where possible, put the affected loops in Manual mode. See TDC maintenance manual.

Error	913
State	Reset.
Description	Optional RAM card read/write failure.
Solution	Where possible, put the affected loops in Manual mode. See TDC maintenance manual.

Error	914
State	Reset.
Description	Controller file power down or low voltage.
Solution	Where possible, put the affected loops in Manual mode. See TDC maintenance manual.

Error	915
State	Reset.
Description	Controller file internal bus failure.
Solution	Where possible, put the affected loops in Manual mode. See TDC maintenance manual.

Error	916
State	Soft Failure.
Description	Errors detected in CPU Control Card.
Solution	Where possible, put the affected loops in Manual mode. See TDC maintenance manual.

Error	917
State	Soft Failure.
Description	Controller file A/D calibration error of 1-5%.
Solution	Where possible, put the affected loops in Manual mode. See TDC maintenance manual.

Error	918
State	Soft Failure.
Description	Point file A/D calibration error of 1-5%.
Solution	Where possible, put the affected loops in Manual mode. See TDC maintenance manual.

Error	919
State	Soft Failure.
Description	RAM battery voltage too low.
Solution	Where possible, put the affected loops in Manual mode. See TDC maintenance manual.

Error	920
State	Soft Failure.
Description	SBLA card C-Link communication error.
Solution	Where possible, put the affected loops in Manual mode. See TDC maintenance manual.

Error	921
State	Soft Failure.
Description	Optional RAM trend data area failure.
Solution	Where possible, put the affected loops in Manual mode. See TDC maintenance manual.

Error	923
State	Soft Failure.
Description	CL/MC program execution overload, exceeded one cycle time.
Solution	Where possible, put the affected loops in Manual mode. See TDC maintenance manual.

Error	924
State	Soft Failure.
Description	Primary CCM is saved (AMC only).
Solution	Where possible, put the affected loops in Manual mode. See TDC maintenance manual.

Error	925
State	Soft Failure.
Description	Primary CCM in standby has failed (AMC only).
Solution	Where possible, put the affected loops in Manual mode. See TDC maintenance manual.

Error	926
State	Soft Failure.
Description	Reserve CCM in standby has failed (AMC only).
Solution	Where possible, put the affected loops in Manual mode. See TDC maintenance manual.

Error	927
State	Soft Failure.
Description	Primary IOM is saved (AMC only).
Solution	Where possible, put the affected loops in Manual mode. See TDC maintenance manual.

Error	928
State	Soft Failure.
Description	Primary IOM in standby has failed (AMC only).
Solution	Where possible, put the affected loops in Manual mode. See TDC maintenance manual.

Error	929
State	Soft Failure.
Description	Reserve IOM in standby has failed (AMC only).
Solution	Where possible, put the affected loops in Manual mode. See TDC maintenance manual.

Error	933
State	Partial Failure.
Description	Controller file A/D calibration point exceeds 5% limit.
Solution	Where possible, put the affected loops in Manual mode. See TDC maintenance manual.

Error	934
State	Partial Failure.
Description	Point file A/D calibration point exceeds 5% limit.
Solution	Where possible, put the affected loops in Manual mode. See TDC maintenance manual.

Error	935
State	Partial Failure.
Description	Controller file output 1 read back error (open circuit).
Solution	Where possible, put the affected loops in Manual mode. See TDC maintenance manual.

Error	936
State	Partial Failure.
Description	Controller file output 2 read back error (open circuit).
Solution	Where possible, put the affected loops in Manual mode. See TDC maintenance manual.

Error	937
State	Partial Failure.
Description	UAC I/F failure.
Solution	Where possible, put the affected loops in Manual mode. See TDC maintenance manual.

Error	938
State	Partial Failure.
Description	Controller file A/D conversion time too slow.
Solution	Where possible, put the affected loops in Manual mode. See TDC maintenance manual.

Error	939
State	Partial Failure.
Description	Point file A/D conversion time too slow.
Solution	Where possible, put the affected loops in Manual mode. See TDC maintenance manual.

Error	942
State	Partial Failure.
Description	No.1 point card file power down.
Solution	Where possible, put the affected loops in Manual mode. See TDC maintenance manual.

Error	943
State	Partial Failure.
Description	No.1 point card file bus failure.
Solution	Where possible, put the affected loops in Manual mode. See TDC maintenance manual.

Error	944
State	Partial Failure.
Description	No.1 point card file access failure.
Solution	Where possible, put the affected loops in Manual mode. See TDC maintenance manual.

Error	945
State	Partial Failure.
Description	No.2 point card file power down.
Solution	Where possible, put the affected loops in Manual mode. See TDC maintenance manual.

Error	946
State	Partial Failure.
Description	No.2 point card file bus failure.
Solution	Where possible, put the affected loops in Manual mode. See TDC maintenance manual.

Error	947
State	Partial Failure.
Description	No.2 point card file access failure.
Solution	Where possible, put the affected loops in Manual mode. See TDC maintenance manual.

Error	948 TO 963
State	Partial Failure.
Description	Point card failure in slot 01 to 16.
Solution	Where possible, put the affected loops in Manual mode. See TDC maintenance manual.

Error	990
State	
Description	Alarm circular list full.
Solution	

Error	991
State	
Description	Sequence circular list full.
Solution	

Error	997
State	
Description	Alarm ACK device changed.
Solution	

Error	998
State	
Description	New sequence master assigned on hiway.
Solution	

UAC primary controller codes

Error	S000
State	Normal.
Description	The Primary Controller is performing the control functions. If the Primary Controller fails a UAC backup will be attempted.
Solution	

Error	Sxx1
State	Hiway Disabled.
Description	Primary Controller disabled by Hiway command.
Solution	

Error	Sxx2
State	Local Disabled.
Description	Primary Controller disabled at the RCD control panel.
Solution	

Error	Sxx3
State	Diagnostic Disabled.
Description	Primary Controller disabled by RCD diagnostics. Refer to controller configuration display error description.
Solution	Service the Primary Controller. See the TDC maintenance manual.

Error	Sx1x
State	Saved (backup).
Description	Primary Controller hard save. The Reserve Controller is performing the control functions.
Solution	Service the Primary Controller. See the TDC maintenance manual.

Error	Sx2x
State	Test Save (backup)
Description	Primary Controller test save. The Reserve Controller is performing the control functions.
Solution	

Error	Sx3x
State	Soft Save (backup)
Description	Primary Controller soft save (EC only). The Reserve Controller is performing the control functions.
Solution	Service the Primary Controller. See the TDC maintenance manual.

Error	S1xx
State	Reset (no backup)
Description	No Reserve Controller available to backup this box. The Primary Controller is not performing control.
Solution	1. Attempt restart (STR command). 2. Where possible, put all affected loops in Manual mode. 3. Service Primary Controller.

UAC reserve controller Director CB (6xx) and EC (8xx) error codes

Error	n40
State	RCD Reset
Description	RCD watch dog timer (WDT) expired. Try to restart.
Solution	Reserve Controller Director (RCD) has failed. See the TDC service manual.

Error	n41
State	RCD Reset
Description	RCD CPU instruction check failure.
Solution	Reserve Controller Director (RCD) has failed. See the TDC service manual.

Error	n42
State	RCD Reset
Description	RCD ROM checksum failure.
Solution	Reserve Controller Director (RCD) has failed. See the TDC service manual.

Error	n43
State	RCD Reset
Description	RCD RAM read/write failure.
Solution	Reserve Controller Director (RCD) has failed. See the TDC service manual.

Error	n44
State	RCD Reset
Description	RCD Transceiver card failure.
Solution	Reserve Controller Director (RCD) has failed. See the TDC service manual.

Error	n45
State	RCD Reset
Description	RCD Decoder card failure.
Solution	Reserve Controller Director (RCD) has failed. See the TDC service manual.

Error	n46
State	RCD Reset
Description	RCD sample time clock (STC) failure.
Solution	Reserve Controller Director (RCD) has failed. See the TDC service manual.

Error	n47
State	RCD Reset
Description	RCD Bus Relay Jumper failure.
Solution	Reserve Controller Director (RCD) has failed. See the TDC service manual.

Error	848
State	RCD Reset
Description	Digital I/O File Control Card failure (for second Controller only).
Solution	Reserve Controller Director (RCD) has failed. See the TDC service manual.

Error	849
State	RCD Reset
Description	Slave File absent (for second Controller only).
Solution	Reserve Controller Director (RCD) has failed. See the TDC service manual.

Error	n56
State	
Description	RCD Sample time clock (STC) Integrity Check failure.
Solution	Reserve Controller Director (RCD) has failed. See the TDC service manual.

Error	xxxSnn
State	
Description	Failure during Replacement Backup, where xxx is the RCD error code and Snn is the box number backed up prior to the failure.
Solution	

UAC reserve controller Director MC error codes

Error	1040
State	RCD Reset
Description	Watch dog timer (WDT) expired. Try to restart.
Solution	Reserve Controller Director (RCD) has failed. See the TDC service manual.

Error	1041
State	RCD Reset
Description	CPU instruction check failure.
Solution	Reserve Controller Director (RCD) has failed. See the TDC service manual.

Error	1042
State	RCD Reset
Description	ROM card checksum failure.
Solution	Reserve Controller Director (RCD) has failed. See the TDC service manual.

Error	1043
State	RCD Reset
Description	RAM card read/write failure.
Solution	Reserve Controller Director (RCD) has failed. See the TDC service manual.

Error	1045
State	RCD Partial Failure
Description	Decoder card failure, incorrect RCD operation mode.
Solution	Reserve Controller Director (RCD) has failed. See the TDC service manual.

Error	1046
State	RCD Reset
Description	Sample time clock (STC) failure.
Solution	Reserve Controller Director (RCD) has failed. See the TDC service manual.

Error	1048
State	RCD Partial Failure
Description	Status output switch card failure.
Solution	Reserve Controller Director (RCD) has failed. See the TDC service manual.

Error	1050
State	RCD Reset
Description	RCD Control card failure.
Solution	Reserve Controller Director (RCD) has failed. See the TDC service manual.

Error	1051
State	RCD Reset
Description	CPU overload or over-run.
Solution	Reserve Controller Director (RCD) has failed. See the TDC service manual.

Error	1053
State	RCD Reset
Description	RCD file bus failure.
Solution	Reserve Controller Director (RCD) has failed. See the TDC service manual.

Error	1054
State	RCD Partial Failure
Description	I/O bus switch control card failure.
Solution	Reserve Controller Director (RCD) has failed. See the TDC service manual.

Error	1055
State	RCD Soft Failure
Description	RCD SBLA card failure.
Solution	Reserve Controller Director (RCD) has failed. See the TDC service manual.

Error	1056
State	RCD STC
Description	Sample time clock (STC) Integrity Check failure.
Solution	Reserve Controller Director (RCD) has failed. See the TDC service manual.

Error	1087
State	RCD Partial Failure
Description	Backup memory file power down or absent.
Solution	Reserve Controller Director (RCD) has failed. See the TDC service manual.

Error	1088
State	RCD Partial Failure
Description	Backup memory file bus failure.
Solution	Reserve Controller Director (RCD) has failed. See the TDC service manual.

Error	1089
State	RCD Partial Failure
Description	Backup memory file access failed.
Solution	Reserve Controller Director (RCD) has failed. See the TDC service manual.

Error	1090
State	RCD Partial Failure
Description	I/O bus switch file power down or absent.
Solution	Reserve Controller Director (RCD) has failed. See the TDC service manual.

Error	1091
State	RCD Partial Failure
Description	I/O bus switch file bus failure.
Solution	Reserve Controller Director (RCD) has failed. See the TDC service manual.

Error	1092
State	RCD Partial Failure
Description	I/O bus switch file access failure.
Solution	Reserve Controller Director (RCD) has failed. See the TDC service manual.

Error	1093
State	RCD Partial Failure
Description	UAC action failure
Solution	Reserve Controller Director (RCD) has failed. See the TDC service manual.

UAC reserve controller CB (6xx) and EC (8xx) error codes reported by the RCD

Error	n60
State	RC Reset
Description	Watch dog timer (WDT) start failure.
Solution	Reserve Controller (RC) has failed. See the TDC service manual.

Error	n61
State	RC Reset
Description	Watch dog timer (WDT) expired and CMOS present.
Solution	Reserve Controller (RC) has failed. See the TDC service manual.

Error	n62
State	RC Fail
Description	Watch dog timer stop failure.
Solution	Reserve Controller (RC) has failed. See the TDC service manual.

Error	n63
State	RC Fail
Description	Output to PV check: relay failure.
Solution	Reserve Controller (RC) has failed. See the TDC service manual.

Error	n64
State	RC Fail
Description	Output to PV check: data failure.
Solution	Reserve Controller (RC) has failed. See the TDC service manual.

Error	n65
State	RC Fail
Description	CMOS remote failure.
Solution	Reserve Controller (RC) has failed. See the TDC service manual.

Error	n66
State	RC Fail
Description	CMOS card absent or write protected
Solution	Reserve Controller (RC) has failed. See the TDC service manual.

Error	n67
State	RC Fail
Description	CMOS local failure.
Solution	Reserve Controller (RC) has failed. See the TDC service manual.

Error	668
State	RC Fail
Description	DMA Transceiver disable failure.
Solution	Reserve Controller (RC) has failed. See the TDC service manual.

Error	669
State	RC Fail
Description	DMA Transceiver enable failure.
Solution	Reserve Controller (RC) has failed. See the TDC service manual.

Error	670
State	RC Fail
Description	DMA Transceiver address failure.
Solution	Reserve Controller (RC) has failed. See the TDC service manual.

Error	868
State	RC Fail
Description	Decoder card disable failure.
Solution	Reserve Controller (RC) has failed. See the TDC service manual.

Error	869
State	RC Fail
Description	Decoder card enable failure.
Solution	Reserve Controller (RC) has failed. See the TDC service manual.

Error	870
State	RC Fail
Description	Decoder card address failure.
Solution	Reserve Controller (RC) has failed. See the TDC service manual.

Error	876
State	RC Fail
Description	Digital I/O junction panel expected but absent.
Solution	Reserve Controller (RC) has failed. See the TDC service manual.

Error	877
State	RC Fail
Description	Incorrect switch operation.
Solution	Reserve Controller (RC) has failed. See the TDC service manual.

Error	878
State	RC Fail
Description	Switch module absent.
Solution	Reserve Controller (RC) has failed. See the TDC service manual.

Error	879
State	RC Fail
Description	Output to PV check: relay failure.
Solution	Reserve Controller (RC) has failed. See the TDC service manual.

Error	880
State	RC Fail
Description	Output to PV check: data failure.
Solution	Reserve Controller (RC) has failed. See the TDC service manual.

Error	881
State	RC Fail
Description	Two switch modules present.
Solution	Reserve Controller (RC) has failed. See the TDC service manual.

UAC reserve controller MC error codes reported by the RCD

Error	1061
State	RC Reset
Description	Watch dog timer (WDT) expired, UAC I/F card present. Try to restart.
Solution	Reserve Controller Director (RCD) has failed. See the TDC service manual.

Error	1063
State	RC Fail
Description	RC valve output to PV check: relay failure.
Solution	Reserve Controller Director (RCD) has failed. See the TDC service manual.

Error	1064
State	RC Fail
Description	RC valve output to PV check: data failure.
Solution	Reserve Controller Director (RCD) has failed. See the TDC service manual.

Error	1068
State	RC Fail
Description	Hiway disable failure.
Solution	Reserve Controller Director (RCD) has failed. See the TDC service manual.

Error	1069
State	RC Fail
Description	Hiway enable failure.
Solution	Reserve Controller Director (RCD) has failed. See the TDC service manual.

Error	1070
State	RC Fail
Description	Hiway address failure.
Solution	Reserve Controller Director (RCD) has failed. See the TDC service manual.

Error	1071
State	RC Fail
Description	C-link address failure.
Solution	Reserve Controller Director (RCD) has failed. See the TDC service manual.

Error	1076
State	RC Fail
Description	Status output junction panel or switch module absent.
Solution	Reserve Controller Director (RCD) has failed. See the TDC service manual.

Error	1077
State	RC Fail
Description	Status output module failure.
Solution	Reserve Controller Director (RCD) has failed. See the TDC service manual.

Error	1079
State	RC Fail
Description	Status output to input check: relay failure.
Solution	Reserve Controller Director (RCD) has failed. See the TDC service manual.

Error	1080
State	RC Fail
Description	Status output to input check: data failure.
Solution	Reserve Controller Director (RCD) has failed. See the TDC service manual.

Error	1081
State	RC Fail
Description	Two status output switch modules present.
Solution	Reserve Controller Director (RCD) has failed. See the TDC service manual.

Error	1082
State	RC Fail
Description	Reserve I/O bus check: connection failure.
Solution	Reserve Controller Director (RCD) has failed. See the TDC service manual.

Error	1083
State	RC Fail
Description	Reserve I/O bus check: bus failure.
Solution	Reserve Controller Director (RCD) has failed. See the TDC service manual.

Error	1084
State	RC Fail
Description	UAC I/F card absent.
Solution	Reserve Controller Director (RCD) has failed. See the TDC service manual.

Error	1085
State	RC Fail
Description	UAC I/F card failure.
Solution	Reserve Controller Director (RCD) has failed. See the TDC service manual.

TDC sequence programs

The Multifunction Controllers can be programmed to execute special control sequences and control strategies. The programs are downloaded to the controllers, monitored and controlled via the Controller Configuration displays; these displays are normally accessed by double clicking on the PV parameter of the associated Point Detail display.

The sequence program states and control actions are described by the TDC Product Manuals, the Supervisor access level is required for all sequence control actions.

The programs can optionally, generate messages that need to be acknowledged or confirmed by an operator. The messages can be directed to a particular operator station or operator based on the Area assignments in the same way as alarms. A Message Indicator, which is next to the Comms Alarm Indicator, flashes green when a new message is received. The Message Summary display can be called up by clicking on the Message Indicator.



Note

The Message Summary display can be used to display messages from sources other than the TDC system, for example, messages associated with alarm conditions.

There are two types of messages:

- Informational—appear in the summary with an “i” icon. When you acknowledge an informational message, it is removed from the summary.
- Confirmable—appear in the summary with a “c” icon. If a Multifunction Controller sequence program has generated the message, the program waits for the operator to confirm the message before continuing. When you acknowledge a confirmable message it changes from a flashing state to a static state. When you acknowledge the message again, it is removed from the summary and the sequence program is allowed to continue.

See the section “CL sequence messages” on page 7 for further information.

System administration

6

This chapter describes how to maintain the TDC Data Hiway configuration and troubleshoot the system. Additional information is available in the *Operators Guide* and TDC Product Manuals listed in the section “Referenced documentation” on page 2.

For information about:	Go to:
Configuration Management	page 2
TDC Box configuration monitoring	page 5
TDC Box configuration backup and restore	page 9
Troubleshooting	page 11

Configuration management

The following table provides an overview of the TDC related configuration together with some suggestions for managing the configuration.



Caution

- 1 To avoid differences between the server database and TDC controller values, changes to the TDC point ranges and alarm limits must be performed using the point detail display in Station. Changes to these values are not written to the controller if they are changed in Quick Builder or using the pntbld utility. Quick Builder or point build files should be used only to create the server database and to backup the database.
- 2 If the TDC controller resident configuration is changed from another system (for example, LCN, BOS, EOS or another server) connected to the same Data Hiway the controller configuration must be repeated on the Experion system to update the information on Station displays.

Configuration Entities	Master Data	Backup Data	Configuration Management	References
DHEB	Primary server .mdb file	Backup server .mdb file	If you have redundant servers, configuration must be repeated on the backup server.	“DHEB software configuration” on page 6
Channel definitions	Primary server	Backup server .qdb file or .hdw file	Use Quick Builder or .hdw file as a backup.	“Defining TDC channels” on page 2
Channel configuration	Primary server	Backup server	Caution: If the channel is rebuilt, the channel configuration information is deleted.	“Channel configuration” on page 6
Controller definitions	Primary server	Backup server .qdb file or .hdw file	Use Quick Builder or .hdw file as a backup.	“Defining TDC controllers” on page 7

Configuration Entities	Master Data	Backup Data	Configuration Management	References
Asset definitions	Primary server	Backup server		“Configuring the Enterprise Model” in the <i>Configuration Guide</i>
Asset assignment	Primary server	Backup server	Caution: If a Station is rebuilt, asset assignments are deleted.	“Configuring the Enterprise Model” in the <i>Configuration Guide</i>
TDC controller I/O slot definitions	TDC controller data	Primary and backup server Backup files	Use the hdwconfig utility to create backup files.	“TDC I/O slot definitions” on page 9 “TDC box configuration backup and restore” on page 9
Point configuration	Primary server	Backup server .qdb file or .pnt file	Caution: See the Caution notice at the start of this topic.	“Defining TDC points” on page 15
TDC controller configuration	TDC controller data	Primary and backup server Backup files	Use the hdwconfig utility to create backup files.	“Additional tasks” on page 39 “TDC box configuration backup and restore” on page 9
Group and trend display configuration	Primary server	Backup server .pnt file	See “Group and Trend display configuration” on page 4	
CL/MC programs	TDC controller data	Primary and backup server .mo files	Maintain backups of the .cl source files.	

Group and Trend display configuration

Points can be added to group displays and trend displays using Quick Builder and Station displays. When a point with a group display or trend display allocation is downloaded to the server from Quick Builder it overwrites the server configuration for that position in the group display or trend display. If multiple points have the same group display allocation, the last point downloaded takes the position in the display. If a point is uploaded from the server to Quick Builder the existing group display allocations, stored in Quick Builder, for the point are deleted. If a point needs to be allocated to several groups (which is normally the case), this can only be achieved by configuring the groups using the Station displays.

To avoid losing the trend and group configuration information, create a point build file that *only* contains the group and trend entries. The point build file can then be used to restore the group and trend information if required.

TDC box configuration monitoring

Configuration monitoring reads up TDC box resident information and compares it with the server image of the box. Parameters which are different will either be tracked (this means they are stored immediately in the image) or alarmed as a mismatch but not tracked. The following tables indicate which parameters are tracked and which are alarmed.

Monitoring is performed when a TDC Controller Configuration display is called up, and by the Exception Scanning depending on the controller type and Hiway Ownership setting when the Exception poll detects a change to a slot's configuration made by another Preferred Access Device.

Table 10 CB box configuration parameters

Tracked on Change		Alarmed on Change
Decimal place	TD	Algorithm
EU zero *	OP HI limit	EU conversion
EU span *	OP LO limit	Temperature units
Ratio	K (gain, multiplier)	Alarm type
Bias	T (integral, lag, multiplier)	PV tracking
Deviation HI limit *	T2 (derivative, lead, bias)	PV initialize
Deviation LO limit *	RV (scale, bias)	Y - input
PV HI limit *		X - input
PV LO limit *		Past mode recall
Integral HI limit		Square root indication
Integral LO limit		Override algorithm parameters

* Tracked into and out of the server point parameter (if built).

Table 11 EC box configuration parameters

Tracked on Change		Alarmed on Change
X - decimal place	OP HI limit	Algorithm
Y - decimal place	OP LO limit	PV tracking
EU zero *	TD	PV initialize
EU span *	K (gain, multiplier)	Square root
Ratio	K1 (integral, lag, multiplier)	Control
Bias	K2 (derivative, lead, bias)	Equation
PV HI HI limit *		Indication
PV HI limit *		TOG interval
PV rate limit *		Past mode recall
PV LO LO limit *		X - input
PV LO limit *		Y - input
Deviation HI limit *		Y conversion
Deviation LO limit *		M - input
Integral HI limit		Alarm type
Integral LO limit		Z - input

* Tracked into and out of the server point parameter (if built). EC SP range not supported; SP will take Range of PV.

Table 12 MFC and AMC box configuration parameters

Tracked on Change	Alarmed on Change
Decimal place*	Algorithm
EU zero*	EU conversion
EU span*	Temperature units
Ratio	PV tracking
PV HI limit*	PV initialize
PV LO limit*	Alarm type
Deviation HI limit*	Square root
Deviation LO limit*	X - input
Integral HI limit	Y - input
SP HI limit	Override control algo
SP LO limit	Override equation algo
OP HI limit	Past mode recall
OP LO limit	Board type
Rate HI limit	Sensor type
Rate LO limit	Input override
TD (digital filter)	Pulse width (DO)
K (gain, multiplier)	Logic blocks
K1 (integral, lag, multiplier)	Indication
K2 (derivative, lead, bias)	
RV (scale, bias)	

* Tracked into and out of the server point parameter (if built).



Note

MFC/AMC configuration data is not tracked while the TDC box is in an IDLE state. This is to avoid tracking of unwanted data. Care, therefore, must be taken when viewing box configuration data while box is in IDLE state.

Table 13 PIU box configuration parameters

Tracked on Change	Alarmed on Change	
Decimal place [*]	Board type ^{**}	Square root
EU zero [*]	Sensor type	Alarm type
EU span [*]	Scan frequency	Pulse width
PV HI limit [*]	Smoothing	Notify suppress
PV LO limit [*]	Significant change	
Deviation HI limit [*]	Temperature units	
Deviation LO limit [*]	Open T/C detect	

^{*} Tracked into and out of the server point record parameter (if built). The appropriate slot change flag is also issued by the server.

^{**} Reset TDC box on data entry.

Table 14 DHP box configuration parameters

Tracked on Change	Alarmed on Change	
Decimal place [*]	Board type	Alarm type
EU zero [*]	Sensor type	Port ^{**}
EU span [*]	Protocol ^{**}	Notify suppress
PV HI limit [*]	Hot start ^{**}	Device address ^{**}
PV LO limit [*]	Significant change	Keep alive address ^{**}
Deviation HI limit [*]	Configured scan time ^{**}	Device ^{**}
Deviation LO limit [*]	Model type ^{**}	Address ^{**}
Actual scan time	Square root	Range ^{**}

^{*} Tracked into and out of the server point record parameter (if built).

^{**} Reset TDC box on data entry.

TDC box configuration backup and restore

The configuration utility, `hdwconfig`, is used to backup and restore the server's image of the TDC box configuration. The utility can also be used to copy the configuration to a different box of the same type or another system.



Note

Before saving the configuration, ensure that the server's configuration image reflects that of the TDC boxes by performing a configuration upload. To upload the configuration the channel and controller must be enabled.

You run **hdwconfig** from a Command Prompt window.

To save the channel or controller configuration data to a file, the syntax is:

```
HDWCONFIG SAVE file {-CHN cc| -RTU rr} [-RENEW]
```

To restore channel or controller configuration data from a file to the specified channel or controller with the same channel and controller, the syntax is:



Note

To restore the configuration to the server image, the system must be running and the controllers must be disabled.

After restoring a configuration file it is sometimes necessary to call up I/O slot definition displays to synchronize the server configuration before the points associated with the I/O slots can be built successfully.

```
HDWCONFIG RESTORE file {-CHN cc| cc-RTU rr} [-FROM nn]
```

To display a list of channels and controllers whose configuration data is contained in the file, the syntax is:

```
HDWCONFIG LIST file
```

Option	Description
SAVE <i>file</i>	Saves the configuration image to <i>file</i> .
RESTORE <i>file</i>	Restores the configuration from <i>file</i> .
LIST <i>file</i>	Displays a list of the channels and controllers contained in <i>file</i> .
-CHN <i>cc</i>	Channel <i>cc</i>
-RTU <i>rr</i>	Controller <i>rr</i>
-RENEW	Forces data for the controller to be saved, even if data already exists in the file.

Option	Description
-FROM <i>nn</i>	Used in conjunction with -RTU <i>nn</i> to restore controllers from <i>nn</i> to <i>rr</i> .

For example, to save controller 21's configuration to a file called box21, type:

hdwconfig save box21 -rtu 21

To save the configuration for all controllers connected to channel 01 to the file hiway01:

hdwconfig save hiway01 -chn 01

To restore the configuration saved as file box21 to the server image for box 21:

hdwconfig restore box21 -rtu 21

To restore the configuration saved as file box21 to the server image for box 22:

hdwconfig restore box21 -rtu 22 -from 21

Troubleshooting

The following is a summary of some of the applications and utility programs that can be used to troubleshoot the system; please see the Station Help menu item for further information.

Application/Utility	Used For	Where to find it	Reference
DHEB Monitor Program	Used to test the communications with between the server and the DHEB	Via the program menu on the primary server. The program is called DHEB Monitor	“DHEB software configuration” on page 6
Diagnostic Capture Tool (DCT)	Used to capture the information required by Honeywell to diagnose system problems	Via the program menu on the primary server	Not applicable
Station Display Page 10 (Channel Scanning Statistics display)	Used to monitoring the load on the scanning subsystem	Select Request Page icon or press F5 and enter the page number 10	Station Help
Server Log file	Log file containing system events and error messages	Via the program menu on the primary server, listed as Experion Server Log	Not applicable
display	Used to show the description for server error codes	Type the command into a Command Prompt Window on the primary server	Station Help
tdctst	Used to test the communications with the TDC Data Hiway. Also used to write to MCs to reset the Sequence Mastership for CL/MC messages	Type the command into a Command Prompt Window on the primary server	“Testing the communications” on page 9
hdwbld, hdwbckbld, pntbld, bckld and listag	Alternative low level configuration tools to Quick Builder	Type the command into a Command Prompt Window on the primary server	Station Help
lissen	Used to analyze and optimize the scanning load	Type the command into a Command Prompt Window on the primary server	Station Help

Application/Utility	Used For	Where to find it	Reference
pntdel	Used to delete all channel, controller, and point configuration data	Type the command into a Command Prompt Window on the primary server	Station Help
alg1st	Lists the free algorithm blocks numbers	Type the command into a Command Prompt Window on the primary server	Station Help
rtusum	Lists all points assigned to a particular controller	Type the command into a Command Prompt Window on the primary server	Station Help
trace	Used to record communications activity for Stations and channels.	Type the command into a Command Prompt Window on the primary server	Station Help
hdwconfig	Used to backup and restore the servers image of the TDC box configuration	Type the command into a Command Prompt Window on the primary server	“TDC box configuration monitoring” on page 5
tdc_cl	Used to compile TDC CL/MC sequence programs	Type the command into a Command Prompt Window on the primary server	“Compiling CL programs” on page 9
prpmon	Used to monitor the point processing performance	Type the command into a Command Prompt Window on the primary server	Not applicable

System monitoring points

The following procedure can be used to create a trend display to monitor the load on the TDC Channel and the Server CPU.

- 1 Add a channel using the User Scan Task Channel type.
- 2 Add a controller to the channel, for example,
Name = SERVERDB
File Number 251 and Record 1
- 3 Add an analog point to monitor the channel data acquisition queue, for example,
Point ID = ZZ_CHN01_DAQ
Description = CHANNEL 1 DATA ACQUISITION QUEUE
PV Source Address, Address Type = FILE
File 20, Record 76, Word 12, Bit 0, Width 16, Format INT2, Controller Name SERVERDB
- 4 Add an analog sinewave point to monitor the server load, an irregular waveform is normally an indication of a Server CPU overload:
Point ID = ZZ_SINEWAVE
Description = SINEWAVE POINT
PV Source Address, Address Type = FILE
File 8, Record 1, Word 114, Bit 0, Width 16, Format INT2, Controller Name SERVERDB
- 5 Add the points to a trend.

Control Language reference

7

This chapter describes how to develop Control Language/Multifunction Controller (CL/MC) programs that run in the Multifunction Controller (MFC) and Advanced Multifunction Controller (AMC). The controllers are functionally identical except that they have a different cycle time (1 second and 0.5 seconds respectively), which can affect the programming. Additional information is available in the TPS and TDC Product Manuals, for example, the *Hiway Gateway CL/MC Manual* and the *SOPL User's Manual*.

The Experion and TPS systems support the CL/MC programming language, which is functionally similar to the SOPL programming language used by the TDC EOS stations. There are some important functional differences, which are mainly associated with the way the Abnormal Handlers/Sequences are executed. Honeywell can provide SOPL to CL conversion services if required.

CL/MC sequence programs and logic block schemes can be used to implement a wide range of continuous and sequential control functions, for example, mass flow calculations, transmitter range switching, special control algorithms, interlocks, sequential and batch control. This chapter assumes a working knowledge of a Multifunction Controller and its internals.

The programs are downloaded to the controllers, monitored and controlled via the Controller Configuration displays that can be accessed by double clicking on the PV parameter of the associated Point Detail display. The Supervisor access level is required for all sequence control actions.

The programs can optionally generate messages that need to be acknowledged or confirmed by an operator. The messages can be directed to a particular Station or operator, based on the Area assignments in the same way as alarms.

For information about:	Go to:
Configuration tasks	page 3
Program development	page 6
Control Language basics	page 16
Statements	page 43

For information about:	Go to:
Structures	page 67
Syntax Summary	page 91

Configuration tasks

CL acronym libraries

When a CL program is compiled, the Sequence Name, Phase Names, Step Labels and words used in the Send statements are automatically saved to a CL Acronym Library resident on the server. The server has 20 CL Acronym Libraries; the TPS HG and TDC EOS stations have 4 libraries. Each MC is assigned to one CL Acronym Library as part of the Controller configuration, see the section “Controller Main tab settings” on page 7. The default allocation for all MCs is library 1. The CL Acronym Library assignment is displayed on the MC Controller Status display. The CL Acronym Library displays can be accessed via the Configuration menu.

Each library consists of 576 strings assigned as follows:

String	Record Number Range
Sequence Names	129 to 256, 513 to 576
Phase Names	129 to 256, 513 to 576
Step Labels	257 to 512
Send String item	1 to 576

If a TPS system or EOS is connected to the same Data Hiway, and these systems also need to monitor or control the CL programs, the HG, EOS and CL Acronym Libraries must be synchronized manually by entering the words in each system library.



Note

If the CL Acronym Library number assignments are changed, the associated CL programs must be recompiled and downloaded to the controllers or the Sequence Names, Phase Names, Step Labels and CL messages will be unreadable because the index numbers will not correspond to the correct text strings.

Process module points

The CL programs are monitored and controlled via points called Process Module Points. Multiple CL programs can be compiled against a single PM Point to allow different programs to be downloaded to the controller for different products or process conditions.

A PM Point must be built for each MC Sequence Slot that will be used. The MCs have 16 Sequence Slots, which allows up to 16 CL program to run concurrently per MC. The PM points should be configured Status Points with the following settings:

Tab	Property	Notes
Main	Point ID	Use a maximum of 10 alphanumeric characters
	Area Code	Determines
	PV Source Address	Controller Name + SEQ nn STATE
	PV Scan Period	5 seconds
	Number of States	8
	State Descriptor 7	ERROR
	State Descriptor 6	FAIL
	State Descriptor 5	PAUSE
	State Descriptor 4	RUN
	State Descriptor 3	END
	State Descriptor 2	LOADED
	State Descriptor 1	DOWNLOAD
	State Descriptor 0	NOT LOADED
Alarms	ERROR	Enabled
	FAIL	Enabled
	PAUSE	
	RUN	
	END	
	LOADED	
	DOWNLOAD	
	NOT LOADED	
	Re-alarm on state transition	Enabled
Control	Source Addresses	Leave blank
	Destination Addresses	Leave blank
	Scan Periods	0

C Link addresses

The CL programs can control the CL programs in another MC and read and write data to another MC if they are connected to the same C Link. The C Link addresses are set via switches on the MC, the server reads the address as part of the box configuration. The CL compiler uses the address when the program is compiled. The C Link address is displayed on the MC Controller Status display, the address can be set via the display to allow programs to be developed without being connected to the Data Hiway. If the C Link address is changed or is incorrect the CL programs must be re-compiled.

Logic blocks

The MC has 128 Logic Blocks that can be used to implement part of the control strategy or used as watchdog timers for the CL programs. When configuring the Logic Blocks, a negative sign can be added to the input to reverse the logic input, a negative sign can also be added to the output which forces the Logic Block to be executed on every cycle. For information on configuring Logic Blocks, see the *SOPL User's Manual*.

Program development

CL software environment

Item	Value
Maximum number of Lines of Code/Sequence Step (depends on statement complexity)	100 to 300
Maximum number of statements per step	254 statements
Maximum blocks of code for each sequence	512 blocks
Block size	16 words
Maximum blocks of code for all 16 sequence blocks slots in MC; the MC must be configured: 'SOPLMEM' = SOPLMEM and 'BOXTREND' = NOTREND	776
Max. phases and steps in a sequence is only limited by the number of slots available for phases and step identifiers in the Acronym Library.	-
Maximum size for expression or condition, where both operators and operands count as 1 item each. Example: $(x + y) < 6$ contains 5 items	100 items
Maximum number of declarations in a sequence (includes locals, externals, and constants). All declarations count as 1 item each.	approx. 270 items
Maximum number of constant declarations in a sequence; only constants that are referenced in the body of the sequence are counted. Declarations for duplicate values are not counted.	256 constant declarations
Words for each statement (depends heavily on complexity of statement) (average is approximately 20 words)	10 words to 400 words

CL program directories

Default Path	Description
C:\Program Files\Honeywell\Experion PKS\server\cl	The CL Compiler must be executed from this directory
C:\Program Files\Honeywell\Experion PKS\server\cl\src	CL source files .cl

Default Path	Description
C:\Program Files\Honeywell\Experion PKS\server\cl\lst	CL list files .ls and .xf generated by the compiler
C:\Program Files\Honeywell\Experion PKS\server\cl\pls	Reference files .d generated by the compiler
C:\Program Files\Honeywell\Experion PKS\server\data\cl	Intermediate object files .mo generated by the compiler

Creating CL programs

Use a text editor to create and edit the CL source files located in the `src` folder. The sequence program consists of a main sequence and optional abnormal condition handlers and subroutines. The abnormal handlers are used to interrupt the main sequence and when predefined conditions occur. The following is a simple example:

```

SEQUENCE CLTEST (POINT MC2_PM02)
  PHASE TESTING (HOLD HLD;SHUTDOWN SHDN;EMERGENCY EMGY)
  L1:      SEND (WAIT):"PLEASE CONFIRM MESSAGE TO CONTINUE"
          SEND: "THIS MESSAGE IS FOR INFORMATION ONLY"
          GOTO L1
END CLTEST
HOLD HANDLER HLD
  SEND : "HOLD HANDLER INVOKED"
  RESTART
  RESUME PHASE TESTING
END HLD
SHUTDOWN HANDLER SHDN
  SEND : "SHUTDOWN HANDLER INVOKED"
  RESTART
  RESUME PHASE TESTING
END SHDN
EMERGENCY HANDLER EMGY
  SEND : "EMERGENCY HANDLER INVOKED PROGRAM END"
  SEND : "MANUAL RESTART REQUIRED"
END EMGY

```

For a list of the parameters that can be accessed by the sequence programs, see the “MC parameters accessible by TDC CL” on page 78 or the appendices in the *SOPL User’s Manual*.

CL sequence messages

The CL programs can optionally, generate messages that need to be acknowledged or confirmed by an operator. The messages can be directed to a

particular operator station or operator based on the Area assignments in the same way as alarms. There are two types of messages:

- Informational— appear in the summary with an “i” icon. When you acknowledge an informational message, it is removed from the summary.
- Confirmable—appear in the summary with a “c” icon. If a Multifunction Controller sequence program has generated the message, the program waits for the operator to confirm the message before continuing. When you acknowledge a confirmable message it changes from a flashing state to a static state. When you acknowledge the message again, it is removed from the summary and the sequence program is allowed to continue.

The Experion system does not support the concept of Sequence Mastership, so changes to these assignments made by an EOS Station or TPS system will prevent the messages from being displayed on the Experion system. In order to reset the owner of the CL messages it is necessary to zero a memory address in the controller using the following procedure:



Caution

Make sure you have a current backup of the box database before performing this command. The action of writing data directly to the controller can corrupt the controller memory which can cause process upsets.

- 1 At the Command prompt type:
tdctst <
- 2 When prompted, enter the Channel Number of the TDC Channel. For example:
chn01 <
- 3 Read the MC address where *bb* is the Hiway Address (box number):
read *bb*,1476 <

The data value returned should be zero, if it is not zero clear the address as follows:

- 1 **write *bb*,1476,0000 <**
- 2 Check the MC address using the read command:
read *bbc*,1476 <
- 3 To quit the **tdctst** utility type:
q <

Check that messages are received and can be acknowledged.

Compiling CL programs

The source file must reside in the `src` folder with the `.cl` file extension (for example, `filename.cl`). The following compiler command must be executed from the `cl` folder not the `src` folder:

```
tdc_cl (filename) [-d -x]
```

Part	Description
<i>filename</i>	The source file without the <code>.cl</code> extension.
<code>-d</code>	Optional switch to compile source lines beginning with <code>%DEBUG</code> see “Embedded compiler directives” on page 66.
<code>-x</code>	Optional switch to cause the compiler to output cross reference information to a file in the <code>lst</code> folder (<code>filename.xf</code>).

This results in a listing file `filename.lst` in the `lst` folder.



Note

The CL Compiler will support will support CL messages with words containing more than 10 characters but this can cause duplicate entries in the CL Acronym Libraries.

Compile time errors

There are three categories of compile time errors:

- Fatal Errors—these errors prevent the compiler from further compilation.
- Errors—which include syntax errors and semantic check errors. Detailed messages are included in the `.lst` list file.
- Warnings—which do not prevent successful compilation. These errors mostly occur as a result of programming omissions and should be studied before proceeding to download the program to the controller.

When the compilation is successful, an `.mo` file is created in the `..\server\data\cl` folder and the number of MC Sequence Memory Blocks is listed at the end of the `.lst` list file in the `..\server\cl\lst` folder, this number is required to allocate the MC memory when multiple sequences are used.

The object code file name has the format `hhbbsppp.mo` where:

Part	Description
<i>hh</i>	Hiway number (channel number) for the MFC/AMC.
<i>bb</i>	Box ID for the MFC/AMC.

Part	Description
<i>s</i>	Sequence slot number of the Process module point for the Sequence.
<i>ppp</i>	TDC CL Acronym Library Record number for the Sequence name If an object file of the same name already exists the compiler gives a message to delete and recompile.

Downloading sequence programs

If multiple sequence programs will be used in the same MC it is important to allocate the Sequence Memory to each Sequence Slot starting with Slot 1. The starting Block Number for Slot 1 is normally 1, the starting Block Number for Slot 2 should be 1 + the size of sequence 1 + spare memory block to allow for program changes. This is repeated for each sequence slot to be used.

It is sometimes necessary to IDLE the controller the first time the sequence programs are downloaded. Thereafter, the sequence programs can be downloaded with the controller in the Processing state if the Sequence Memory allocations are not changed.

To download the sequence program:

- 1 Go to the point detail display for the associated Process Module Point.
- 2 Double click the PV to call up the Controller Configuration display for the Sequence Slot.
- 3 Check that the Unit State is Off.
- 4 Select Download Sequence then enter the Sequence Name or the last 3 digits of the .mo file name.
- 5 Enter the starting Memory Block Number followed by enter.
- 6 After the sequence has been downloaded, change the Unit State to Ready then Start the sequence.

See the *TPS CL/MC Manual* or *SOPL User's Manual* for information on the Unit and Sequence States.

Troubleshooting sequence programs

The current state and position of the sequence program can be determined from the Controller Configuration display for the Sequence Slot. The display lists the current Phase Name, Step Name and Statement Number being executed by the main sequence, subroutines and abnormal handlers. The execution path can be traced by comparing the display with the .lst list file generated by the compiler. The LOC field in the listing is the Statement Number.

The sequences can be executed in Single Step, Semi-Auto and Auto modes and Pause statements can be added to the programs to aid debugging. The sequence can be stepped forward or backward Phase, Step or Statement.

The sequences execute a lot faster than the display refresh rate so press the escape key (Esc) to refresh the display as fast as possible, for example every second. Sequence program that loop will appear to have stopped because a loop can be executed at 0.5 second intervals (for an AMC, 1 second for the MFC). The

program execution pauses at the Preemption Points, which are Phase and Step statements plus backward branches.

See the *TPS CL/MC Manual* or *SOPL User's Manual* for further information.

**Note**

The Digital Input Override facility is not fully supported the Experion system.

Sequence program run time errors

Run time errors are typically caused by divide by zero errors and calculation overflows, and attempts to access an I/O signal which has failed, unless the program contains the logic to detect this condition.

Uploading sequence programs

If required, sequence programs can be uploaded from the controller and saved on the server in the `..\server\data\c1` folder. If a file with the same name already exists in this folder it will be overwritten. The facility can be used to backup the object files developed on another system so that they can be downloaded from the server.

Comparison with the TPS CL/MC compiler

This TDC CL compiler is different to the LCN CL compiler. This section describes the main differences and is only included as a comparison for those familiar with the LCN implementation.

Compiler usage

File handling

The filenames are case-sensitive. The standard filename extensions used by the compiler are also case-sensitive, and are set to be in lower case. That means, the source programs must have an extension of .c1 and not .CL. Also the filename specified in the argument should be in correct case.

Command-line arguments

The only TDC CL arguments are -x and -d.

Compiler directives

The TDC CL compiler supports the %DEBUG as is supported by the LCN version. However, %PAGE directive is not supported. If included in a source program %PAGE directive is ignored by the compiler.

Language specification

This TDC CL compiler supports only CL/MC functionality. The *Control Language Reference Manual* PC11-285 dated March 1989 has been used as the base reference. This section describes the differences in implementation of this TDC CL compiler over the specified base.

Identifier

All identifiers in this TDC CL can be 10 characters long, instead of 8 characters for LCN version. Also, the current implementation does not reject identifiers with consecutive break characters. However, an identifier can not start with a break character.

Reserved words

Most reserved words are same as for LCN CL/MC. However the following words, which are predefined identifiers in LCN, are RESERVED in this implementation:

ABS
LOGICAL
NUMBER
OFF

ON

Constants & constant expressions

All constant expressions (that is, expressions involving numerical constants or pre-declared local constants) are evaluated to a single constant at compile time. The resulting value is also range checked. An error is reported for out of range values.

The LCN compiler does not perform constant expression evaluation, or range checking.

TDC CL statements

Syntax of most statements remains unchanged except for the following:

- LOOP statement

The syntax of LOOP and REPEAT statements is redefined below. Independent and REPEAT statements do not exist any more. REPEAT statement acts as the TERMINATOR of the LOOP. Nesting of LOOPS is allowed. However, inner loops must be fully enclosed within the external loop.

ALLOWED	NOT ALLOWED
LAB1: LOOP	LAB1: LOOP
--	--
--	--
LAB2: LOOP LAB2: LOOP	
--	--
--	--
REPEAT LAB2 REPEAT LAB1	
--	--
--	--
REPEAT LAB1 REPEAT LAB2	
--	--
--	--

- Send statement

The Advanced control point (ACP) Send statement is not supported.

MC variable parameters

The MODATTR and NMODATTR parameters for LP, AO and DO type variables are supported by the compiler in order to maintain compatibility with the TPS LCN. However, the system displays do not support these parameters.

Compiler operation

This TDC CL Compiler is a two pass compiler. First pass scans the source program for declaration statements to collect user defined SYMBOLS, Program UNITS, Type definitions, Subroutine arguments etc. Second pass scans the source

file again, in association with data collected by first pass. This time actual code is generated. If compilation is error free, the generated code is also saved in the object file in proper format. The program listing is also generated in this pass. However, errors detected in first pass are listed in the first pass itself.

There is a minor third pass of the compiler. This pass only sorts the generated listing file.

The listing file generated by the compiler has the filename extension of `.lst` irrespective of errors or no errors. This file is stored in the `lst` folder in the current folder.

Control Language basics

This topic introduces you to the fundamental building blocks of TDC Control Language (TDC CL). As with any language, TDC CL has characteristics that enable you to do certain things, while not allowing you to do others. By comparing the characteristics of the English language with characteristics of TDC CL, there are analogies that can help you to understand TDC CL.

This topic contains the following major sub-topics:

- Rules and elements
- Data types
- Variables and declarations
- Expressions and conditions

The description of each TDC CL element, statement, and structure is presented in a four-part format, as follows:

Table 15 Description Structure of TDC CL Elements

Part	Description
Definition	A brief “what is it” discussion.
Syntax	Elements, statements, and structures are built following a specific form; the form specification is called syntax. Another word for syntax is grammar, as previously mentioned. The form of anything you want to build in TDC CL must exactly follow the syntax, so that your structure can be compiled without syntax errors. In “Syntax” on page 18 the way in which the syntax for an element, statement or structure is presented is discussed. See “Syntax summary” on page 91 for a syntax summary for all elements, statements, and structures covered in this manual.
Description	Applies to most, but not all elements, statements, or structures; a description explains non-obvious attributes in more detail; for example, complex syntax or any restrictions that may apply.
Example	Contains typical uses of the elements, statements, or structure, with incorrect uses included for contrast.

Rules and elements

This section describes the rules and basic elements (or building blocks) that are used when building complex elements (Data Types, Variables and Declarations, Expressions and Conditions), or TDC CL Statements and Structures.

Character set definition

The TDC CL character set is composed of the 95 printable characters (including blank) of ASCII. TDC CL is not character case-sensitive (that is, A and a are interchangeable).

Compatibility with the ISO 646 standard is discussed in “TDC CL restrictions” on page 19.

Characters can be combined to generate the following basic elements:

- Comments
- Identifiers (including reserved words; see under “Identifiers” on page 20)
- Numbers
- Quotes
- Special symbols (such as +, -, /, *)

Basic elements are further combined to produce complex elements such as data types, variables and declarations, and expressions and conditions. The complex elements are then used within statements (“Statements” on page 43) and the statements are combined to build structures (“Structures” on page 67).

Spacing

Adjacent elements can be separated by any number of spaces. Spaces are not required between elements except to prevent confusion (that is, at least one space must separate adjacent identifiers or numbers). Spaces cannot appear within an element other than in quotes and comments.

Lines

Your structure, whether you write it out on paper first or enter it into a file using a Text editor, consists of a sequence of lines. Lines are also called source lines; the source code is what the compiler uses to create executable object code. The following applies to the construction of source lines.

No basic element can overlap the end of a source line. Complex elements can continue onto succeeding source lines.

A statement (see “Statements” on page 43) can be continued onto successive lines. Each continuation line must have the ampersand character (&) in its first column. The end of the preceding line and the continuation character are treated as spaces.

A statement can start at any column on a line, subject to the restrictions stated in this section. Indentation is optional; see the sample structures in “Structures” on page 67 for an idea of what good indentation techniques can do for the readability of a structure.

Each statement must begin on a new line, unless it is embedded in another statement (such as IF, ELSE, or a WHEN ERROR clause).

Lines may be blank. Blank lines and comment lines cannot appear between continuation lines. The following examples show valid use of continuation lines:

```
IF hi > foo THEN (SET foo = hi;
& GOTO jail)

ELSE SET foo = low --NOTE: no continuation
jail:
& EXIT
END good
```

The following examples are all invalid

```
IF fred > 3 THEN (SET x = y;
    GOTO gaol) -- must use continuation here
END bad
```

Syntax

Because the syntax definition is part of the discussion of most elements, statements, and structures, you should become familiar with how the syntax is presented.

The syntax for each element, statement, and structure is presented along with its discussion in diagram form. The diagram is entered on the left side with the item you want to build in reverse-video/bold lettering. Follow the arrows to build the item, looping back optionally or when necessary (for example, to build an ID, you must loop back through letter or digit or as many times as required to produce the ID string), until you exit the diagram on the right.

Syntax diagrams have three different symbols with text in them: rectangles, rectangles with curved ends, and circles. Items in rectangles refer to a diagram in another part of the manual. Rectangles with curved ends are reserved words in TDC CL and must be written exactly as shown. Items in circles are usually arithmetic operators and delimiters that must be included exactly as shown when building a complex item such as an expression or a TDC CL statement.

A summary of TDC CL syntax for all elements, statements, and structures in both diagram form and in BNF is found in “Syntax summary” on page 91.

TDC CL restrictions

This section lists the restrictions placed on the language.

Length of identifiers

The following restrictions apply to identifiers (see “Identifiers” on page 20):

- Data point identifiers can be no longer than sixteen characters.
- Parameter identifiers can be no longer than ten characters.
- Sequence program identifiers can be no longer than ten.
- Messages sent to the server Station can be no longer than 60 characters.
- Enumeration-state identifiers can be no longer than ten characters.
- Enumeration type identifiers can be no longer than ten characters, must begin with an alphabetic character, and cannot contain ISO 646 foreign characters.
- Words in messages must be separated by blanks.
- Words can be a maximum of ten characters long. There can be a maximum of seven words in a message.

ISO 646 Compatibility The TDC CL character set is compatible with the international standard ISO 646 character set, of which ASCII is a variant. Certain character positions in the ISO 646 character set are permitted to vary for national use (see “Variable Characters in ISO 646” on page 19). Of these, TDC CL uses only the dollar sign.

Several national variants of ISO 646 use character positions 4/0, 5/11 through 5/14, 6/0 and 7/11 through 7/14 as alphabet extensions, for example, accented or umlauted characters. Such characters cannot be used as alphabets in TDC CL. (They can be used in Strings and comments.)

Table 16 Variable Characters in ISO 646

Position	ASCII	Comments
2/3	#	also pound-sterling symbol
2/4	\$	also currency symbol
4/0	@	varies
5/11	[	varies
5/12	\	varies
5/13	]	varies

Table 16 Variable Characters in ISO 646 (*continued*)

Position	ASCII	Comments
5/14	^	varies sometimes
6/0	‘	varies sometimes
7/11	{	varies
7/12		varies
7/13	}	varies
7/14	~	varies sometimes; also overline

When the characters comma (,), quotation mark (“), apostrophe (’), grave accent (À), or upward arrowhead (^) are preceded or followed by a backspace, ISO 646 prescribes that they be treated as diacritical signs (for example, accents); however, TDC CL does not respect this use. The grave accent and upward-arrowhead characters are not permitted outside of strings and comments.

Comments

A comment begins with a double hyphen (--) and is terminated by the end of the source line. Comments can be seen in examples throughout this manual. A comment is not continued onto a continuation line, but a new comment can appear on each of several continuation lines. Position ASCII Comments.

Correct examples of comments

```
--A long introductory comment should be
--written like this, with a
--separate “-” on each line.
```

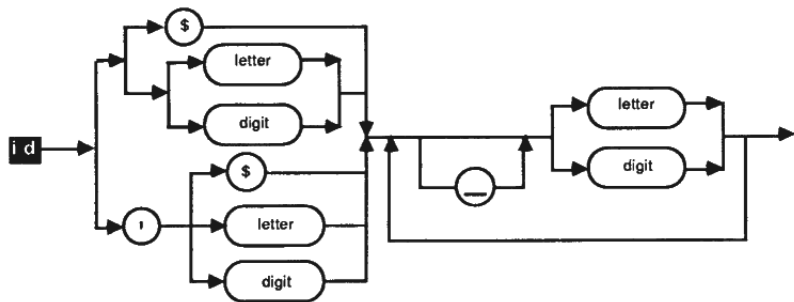
Incorrect examples of comments

```
--This is a long comment such
--as you might use & to prefix a subroutine. It is faulty
--because a comment cannot span
--source lines.
```

Identifiers

Identifiers are used as the names of all kinds of objects in TDC CL: variables and constants, data types, program labels, data point names, etc. The keywords of TDC CL are also identifiers.

Syntax



Identifiers description

Identifiers are composed of the dollar sign (\$), alphabetic characters (A to Z, and a to z.), numeric characters (0 to 9), and the break character (underscore or underbar (_)). Special identifiers (see “Special identifiers” on page 23) are prefixed with an apostrophe ('), but the apostrophe is not part of the identifier. An identifier (except for special identifiers) cannot be entirely numeric; that is, every identifier must contain at least one dollar sign, alphabetic, or break character.

Break characters are used to divide a long identifier so that it can be more easily read. A break character cannot be the first or last character of an identifier.

Upper-case and lower-case characters can be used in identifiers, but the distinction between cases is not significant; that is, each lower-case character is considered the same as its upper-case counterpart. Within the restriction that no basic element may overlap the end of a source line, an identifier can be of any length (except as noted in “Length of identifiers” on page 19).

Any identifier can begin with a dollar sign. This permits use of Honeywell-supplied standard identifiers and other objects whose names begin with dollar signs. Within a TDC CL program, there is no restriction on using identifiers that begin with dollar signs; however, The TDC CL compiler does not let you create any new object that is visible outside the program (as, for example, the sequence program name).

Identifiers examples

VALVE	--	a valid identifier
valve	--	the same as VALVE
Valve	--	same as VALVE and valve
hot_pot	--	a valid identifier
hotpot	--	NOT the same as hot_pot
hot_pot_	--	NOT VALID (trailing break)

```

pump2      --      a valid identifier
2N1401     --      also a valid identifier
14_34_6    --      also valid
14346      --      NOT VALID (all numeric)
$abc       --      valid identifier, restricted use
$4995      --      also valid
a$bc       --      NOT VALID (dollar sign not leading)

```

Box data point identifiers

A box data point is a data point associated with a process-connected box, such as a Multifunction Controller (MFC). It represents box parameters that are visible to TPS system components, including TDC CL programs that are running in the box. These parameters can include internal variables of the box.

Box data-point identifiers follow a naming convention that establishes a set of names of the format:

`$HY nn B mm`

where

nn is the Hiway number *mm* is the box number on the Hiway
mm is the box number on the Hiway

Example `TM(03).STATE` -- internal (MFC) view of timer state
 `$HY01B25.TM(03).STATE` -- one MFC accessing a timer's state
 -- in a different MFC through the C-LINK

Reserved words

“Reserved Words” on page 22 shows a list of the TDC CL reserved words (identifiers) that cannot be redefined by any structure.

Table 17 Reserved Words

ABORT	FOR	NOT	SEND
ABS	FROM	NUMBER	SEQUENCE
ALARM	GOTO	OR	SET
AND	HANDLER	OTHERS	SHUTDOWN
ARRAY	HOLD	OUT	STEP
AT	IF	PAUSE	SUBROUTINE
CALL	IN	PHASE	THEN
EMERGENCY	INITIATE	POINT	WAIT

Table 17 Reserved Words

END	LOCAL	READ	WHEN
ERROR	LOGICAL	REPEAT	WRITE
EXIT	LOOP	RESTART	XOR
EXTERNAL	MINS	RESUME	
FAIL	MOD	SECS	

There are also a number of predefined identifiers in TDC CL. These identifiers are not reserved and can be redefined in a program.

We recommend that you avoid redefining any predefined identifiers, except when the result would not be confusing.

Predefined identifiers are type names, state names of predefined discrete types, insertion point names, and built-in function and subroutine names.

Predefined Discrete types-the following type and its states is predefined:

`Logical = Off/On`

The following is a list of all the predefined identifiers in alphabetical order.

<code>Day_Time</code>	Built-in function
<code>Exp</code>	Built-in function
<code>Int</code>	Built-in function
<code>Ln</code>	Built-in function
<code>Log10</code>	Built-in function
<code>Max</code>	Built-in function
<code>Min</code>	Built-in function
<code>Round</code>	Built-in function
<code>Set_Time</code>	Built-in subroutine
<code>Sqrt</code>	Built-in function

Special identifiers

If an identifier is directly preceded by an apostrophe ('), that identifier is treated as an identifier, even though it may be spelled the same as a reserved word or is all numeric. There must be no spaces between the apostrophe and the identifier.

Except for conflict with reserved words or numbers, a special identifier must follow all the usual rules. The apostrophe cannot make a bad identifier good.

Example	EXTERNAL 750102	-- invalid, 750102 is a number
	EXTERNAL '750102	-- OK, 750102 is an identifier

Conflicts between identifiers

Under some circumstances, the same identifier can be used to name more than one thing without conflict. Under other circumstances, an attempt to reuse an identifier can cause a compile-time error.

The rules under which an identifier can safely name more than one thing are as follows.

- 1 There are four groups of identifiers that can be named: data types, objects, and program units. Two identifiers from the same group cannot have the same name if they are visible in the same scope (see rule 2 for a discussion of scopes.) The identifier group can always be distinguished by the compiler, so a datatype name and a program-unit name (for example) can never cause a conflict, even if they use the same identifier.

- Data Types are:
 - Number
 - Logical
 - Enumerations
- Objects are:
 - Local variables
 - Built-in functions
 - Arguments
 - Data points
 - Parameters
 - Enumeration states
- Program Units are:
 - Phases
 - Steps
 - Labels
 - Sequence programs
 - Abnormal condition handlers
 - Subroutines

Subroutines are program units, but Functions are objects. This means that a Subroutine can have the same name as a local variable, but a Function cannot.

2 Identifiers do not cause conflict if they are declared in different scopes.

- Scopes are:
 - Sequence Programs
 - Abnormal Condition Handlers
 - Subroutines
 - Phases
 - Steps

Most things can be declared in only a few of these scopes. For example, a Step can contain only label declarations.

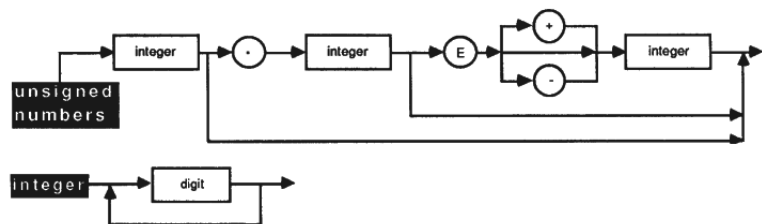
Scopes can sometimes be nested. A Step must be within a Phase, and a Phase within a Sequence Program. Local Subroutines of Sequence Program are considered to be nested within that program unit.

When two scopes are nested, an identifier in an inner scope hides anything in an outer scope that has the same identifier and the same class (Program Unit, Data Type, or Object). For example, a subroutine argument called X hides any local variable called X in the main program.

Numbers

A Number (or Numeric literal) is an ordinary decimal number, with or without decimal point, and with an optional exponent.

Syntax



Description

Numbers that contain a decimal point must have at least one digit both to the left and to the right of the decimal point.

A Number that contains a decimal point can also have an exponent. An exponent consists of the letter E (either upper case or lower case), optionally followed by a plus or minus sign, followed by one or more digits.

A Number cannot contain spaces or break characters. In particular, spaces between a numeric literal and its exponent are not permitted.

Numbers examples

```

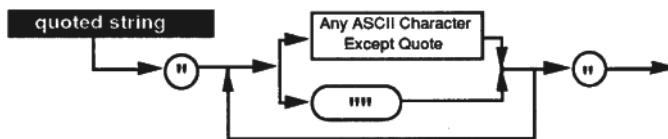
1000      -- valid
1000.     -- NOT VALID; no trailing digit
1000.0    -- valid; same as 1000
0.5       -- valid
.5        -- NOT VALID; no leading digit
10.02E1   -- valid
10.02e1   -- valid; same as 10.02E1
10.02 e1  -- NOT VALID; embedded space
1234.0E-2 -- valid
1234E-2   -- NOT VALID; "1234E" is an identifier
1234.0E -2 -- NOT VALID; embedded space
1234.0E+2 -- valid

```

Quoted string

A quoted String is a sequence of zero or more characters enclosed at each end by quotation marks (“”).

Syntax



Description

Any printable character can appear in a quoted String (string of legal TDC CL characters). If a quotation mark appears in a quoted String, it must be written twice.

In TDC CL, quoted strings are used in only the SEND statement (see “SEND statement” on page 59).

Subject to the restriction that no basic element can overlap the end of a line of source text, a quoted String can be of any length; however, individual words within a quoted string are limited to ten characters.

Quoted string examples

"This is a String" -- a quoted String
 "&\$?*! system" -- can contain any characters
 "He said ""hello"" -- he said "hello"
 "A" " " """" -- three strings of length 1

Special symbols

The characters and combinations of characters in "Special symbols" on page 27 are special symbols.

Table 18 Special symbols

Symbol	Meaning
+ - * / **	Arithmetic operators
< <= > >= <>	Relational operators
=	Equality, assignment operator
..	Range separator
()	Parentheses
: ; ,	Punctuation
"	String separator
-	Comment separator
&	Line continuation
'	Special identifier prefix

Data types

This section describes the types of data that TDC CL can manipulate.

Two kinds of data types are built into the language: scalar and composite.

Scalar types have no components. They are the built-in types: Number, Time, and Discrete types (Logical and Enumeration types).

Composite types are data points (because a data point can have lots of parameters/components), arrays (again, lots of components), and the built-in type String.

Number data types

All numeric values in TDC CL are of the single type, Number. This type is conceptually a subset of the real numbers and is internally implemented in floating point.

There is no separate Integer type in TDC CL; numbers may, of course, have integer values, and TDC CL built-in functions support truncation and rounding of non-integer values.

Numeric data point parameters are represented in single-precision floating point.

Time data type

The only use of Time values in TDC CL are in the WAIT statement, the PHASE heading’s ALARM clause, and the built-in subroutine Set_time. See “Structures” on page 67 for details.

The minimum time resolution is one second, and non integer time expressions are truncated to the next lower integer value (for example, 2/3 MINS becomes 0 minutes). Time expressions can be in MINS or SECS but not both.

Example	Local sked at NN(1)
	...
	SET sked = 20 -- set a numeric variable
	WAIT (sked-3) SECS -- wait 17 seconds
	WAIT (sked/60) MINS -- wait 0 minutes

Discrete data types

Real-world values are either continuous or discrete. Continuous values are often called analog and discrete values digital, because of the type of electronic circuitry used to bring these values into and out of a computer, but they are all represented in digital form in the computer.

TDC CL expresses all continuous values with the type, Number. Discrete values are expressed by the type Logical, and by Enumeration types.

A discrete type has two states. Each state has a name and is distinct from all other states. The order in which the states are named in the type declaration is significant. This means that two discrete types that have the same state names can be different types, because the order in which the states were declared differed. For example, green/blue is different from blue/green.

Variables of discrete types can be compared or assigned to only variables or values of the same type.

Shared state names

A state name can be used in more than one discrete type. For instance, there might be a discrete type whose states are open and close and another whose states are open and shut. Although the respective open states in this example have the same name, they are not the same state. They cannot be compared, and a value of one type cannot be stored into a variable of the other type.

Enumeration types

Many Enumeration types are predefined in a TPS System. These appear just as if those types had been defined in TDC CL and compiled into the system data base at some earlier time.

A few Enumeration types are known to the TDC CL compiler and are predefined by the compiler, rather than by the system; however, there is no visible difference between a compiler-defined type (that is, the predefined discrete-type Logical) and system-defined type with states Off/On.

Variables of Enumeration Types can be declared in TDC CL programs. Like all variables, these can be assigned only values of their type; thus, a variable of Enumeration type Red\Blue can be assigned only one of the values Red and Blue. The value Green cannot be assigned to such a variable.

The only operations defined on Enumeration types are assignment and comparison for equality and inequality.

Logical type

Logical is a predefined Discrete type that has two states: on and off. Unlike Enumeration types, the following operations are defined on Logical values: AND, OR, XOR, and NOT.

Logical should not be considered the same as Pascal's Boolean type, or FORTRAN's LOGICAL type, because it is intended to only represent the state of a discrete variable. It does not represent truth or falsity. In TDC CL, truth values are found in only conditional tests and cannot be stored in variables. If you are

familiar with Pascal, the comparison of program fragments in “Pascal Boolean vs. TDC CL Logic” on page 30 should show this difference.

Table 19 Pascal Boolean vs. TDC CL Logic

Language	Pascal	TDC CL
	VAR flag:Boolean	LOCAL flag:Logical
	X:real LOCAL	x:Number
Method		
A	flag :=x<5	SET flag=(WHEN x<5 : On; &WHEN x>=5 : Off)
BB	IF x<5	IF x<5THEN SET flag=On
	THEN flag:=true	ELSE SET flag=Off
	ELSE flag:=false	

In Pascal method A, the result of the comparison $x < 5$ is considered to be a value and is stored in the variable flag. Engineers with limited programming expertise find this is hard to read and understand. TDC CL method A is just as efficient as Pascal method A and is easier to read.

Pascal method B and TDC CL method B are the same. They are both easy to read, but each is a little less efficient than method A (assuming everything else is equal).

Data points data type

Data points are named composite structures that have named components called parameters. Data points are defined by the Point Builder, rather than by TDC CL. A data point is like a Pascal RECORD. Parameters are identified by dot notation; that is, the Point name is followed by a dot, which is then followed by the parameter name.

Example	A100.PV -- parameter PV of data point A100

Bound data point

Every TDC CL program must be bound to a particular Data point in order to execute. Sequence programs (for discontinuous control) are bound to process module data points. The bound Data point is identified in the heading of the TDC CL program to which it is to be bound, by specifying the Point ID.

TDC CL programs can refer to Data points other than the bound data point, as long as those other data points are declared in external declarations, or are indirectly referenced. Parameters of such Points are accessed by dot notation, as previously described; however, parameters of the bound Data point are directly referred to, without dot notation. It is an error to reference parameters of the bound Data point by dot notation.

Aliasing

Aliasing occurs when the same datum is accessed through different names. Aliasing may arise in TDC CL because data points can be referred to in several ways: directly (by being named in an EXTERNAL declaration) or implicitly (the bound Data point).

TDC CL forbids aliasing. No program should be bound to any Data point that is declared in an EXTERNAL declaration, or to which the program indirectly refers.

Arrays data type

In the MFC, arrays can be of only type Logical or Number, and can have only one dimension. The index type must be Number.

An array whose index type is Number can be indexed by any arithmetic expression. If the result of the index expression is not an integer, it is truncated to the next integer.

Example	<code>coeff (6) -- one dimension</code> <code>pump(x+y*z) -- indexed by expression</code>

String data type

String variables are not supported; string literals (quoted strings) are allowed in only the SEND statement.

Variables and declarations

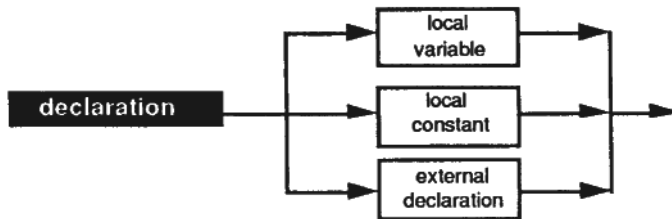
Overview

This section describes the declaration, use, and scope of functions, local constants, and local, external, and parameter variables.

Local variables are owned and defined by a TDC CL program. External variables are data points that are defined outside TDC CL. Parameter variables are parameters of the bound data point or of some external variable.

All declarations must precede any executable statements in their program. In addition, function definitions must follow the declarations of any variables that they use.

Syntax

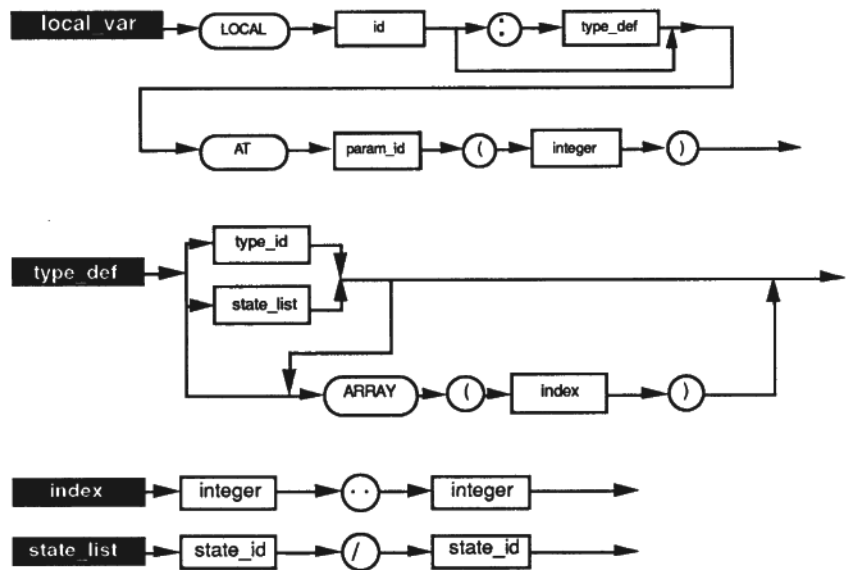


Local variables

Local variables are declared in the heading of a sequence program. The local variable is an MFC numeric or flag, and therefore exists outside the sequence.

Local variables are visible to only the program they are declared in and to any local subroutines belonging to that program.

Syntax



Local variables description

LOCAL variables are of only type number (default), logical, or 2-state enumeration. The Enumeration type can be located in only a flag variable and is restricted to either a state name list containing two states or the predefined enumeration types used for MFC Data point parameters that have two states.

One-dimensional arrays of LOCAL variables can be specified. The array index must be a number declared by naming lower and upper bounds. The lower bound is the left-most integer in the index. The upper bound is the right-most integer in the index and its value must be greater than that of the lower bound.

The AT clause defines the location of the local variable and must name one of the MFC's numeric (NN) or flag (FL) variables, using its box `data_point_` parameter identifier; the compiler places the local variable at that location. If the local variable is an array, its first element is placed at the named location and the remainder of its elements occupy contiguous variables in ascending order from that named.

Note that when a type other than logical is given to a local variable, it is no longer compatible with logical, even though its hardware location may be a flag location. When the variable is used in a SEND statement, the values are displayed in terms of the state names of the type in the declaration.

Local variables examples

```
LOCAL mu: Logical AT FL(12)      -- MFC flag variable
LOCAL ka AT NN(04)              -- MFC numeric variable
LOCAL jo: ARRAY(5..9) AT NN(80) -- MFC numeric array
LOCAL temp1 : open/close AT FL(01) -- State name lists and
LOCAL temp2 : STATE AT FL(02)    -- enumeration-type IDs in
-- local declarations
```

Local constants

Local constants of the data type Number can be declared.

Syntax



Description

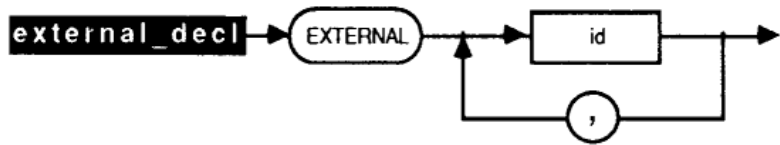
Local constants cannot be modified.
For TDC CL constant expressions must be composed of arithmetic operators, the built-in function ABS, numeric literals, and identifiers that have been previously declared as LOCAL constants. No other functions are permitted.

Example	LOCAL pi = 3.14159265	-- a numeric constant
	LOCAL ten_K = 1.0e4	-- another
	LOCAL 2_pi = pi * 2	-- a constant expression

External data points

Data points other than the bound Data point can be accessed by a TDC CL program only if they are named in an EXTERNAL declaration, or if they are indirectly accessed.

Syntax



Description

The EXTERNAL declaration introduces the name of one or more data points. None of these can be the bound Data point. Each external Data point must already exist in the system at the time the program is compiled; otherwise, the compiler reports an error. External Data points can be declared in sequence programs, but *not in subroutines*.

EXTERNAL declarations can name only MFC data points (including points in other MFCs on the same C Link). Data points in the HG and other LCN modules cannot be referenced.

Example	EXTERNAL anp049hx, AX_001 EXTERNAL \$HY03B22 --- Box Data Point

Process modules definition

A Process module is a Data point that is used as the platform for sequence execution. Each sequence program, before it can be executed, is bound (loaded) to a specific Process module. The Process module is the bound Data point of a sequence program. Refer also to “External data points” on page 34.

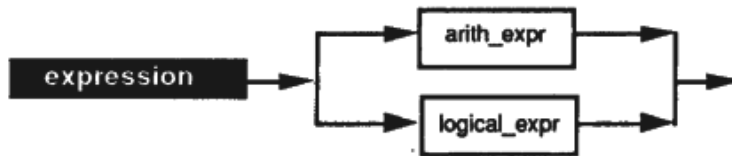
Certain parameters of the related box Data point are made to appear (to the TDC CL compiler) as parameters of the process module Data point. This enables convenient access to box numerics, timers, flags, I/O slots, and controller slots.

Expressions and conditions

Overview

This section describes the formation of arithmetic expressions, Logical expressions, and conditions, which are an extended variety of Logical expressions used in conditional tests.

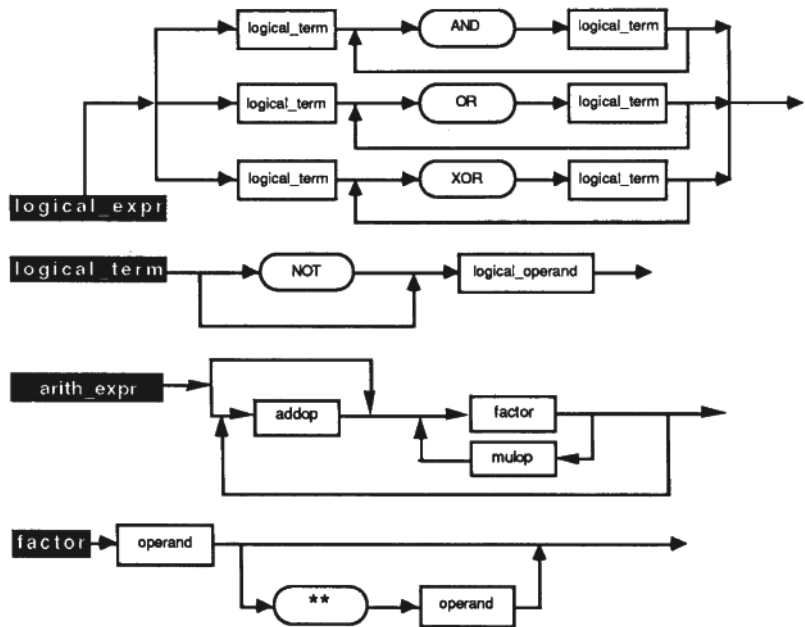
Expressions & conditions syntax



Arithmetic & logical expressions

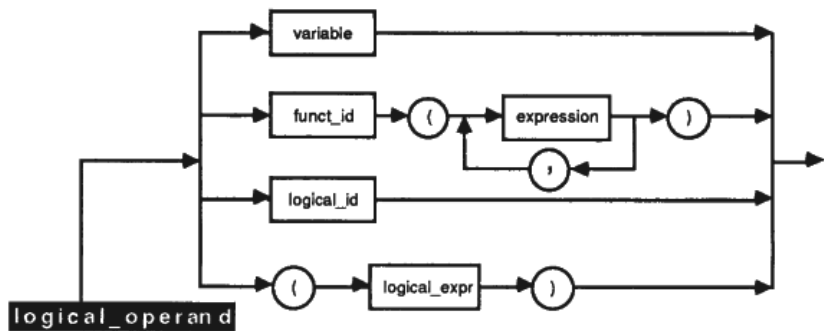
An expression is a formula that defines the computation of a value. The components of an expression are operands and operators.

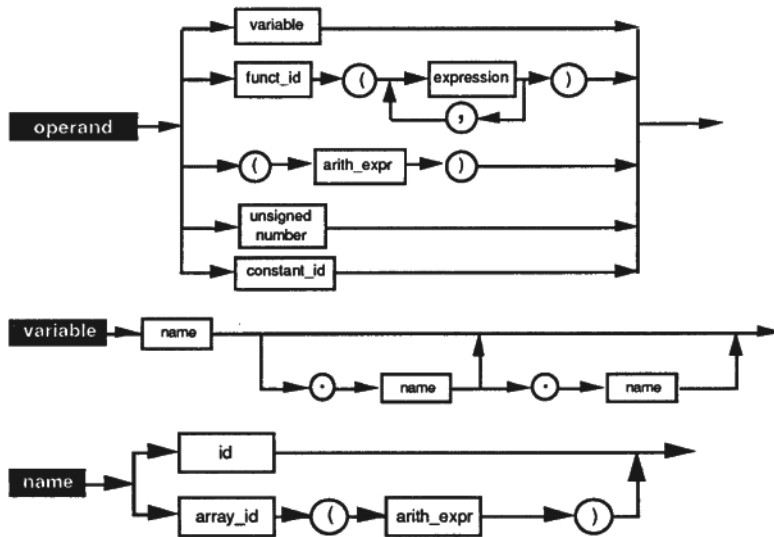
Syntax



Operand definition

An operand can be a variable, a parameter of a Data point, an array element, a number, a quoted String, a discrete-state identifier, a function call, or an expression enclosed in parentheses.





Operators definition

Operators operate on one or two operands and produce a value that can itself be operated on. Operators can be monadic (take a single operand) or dyadic (take two operands). Operators bind according to the priority order given in “Operator Priorities” on page 38; higher priority means closer binding.

Table 20 Operator Priorities

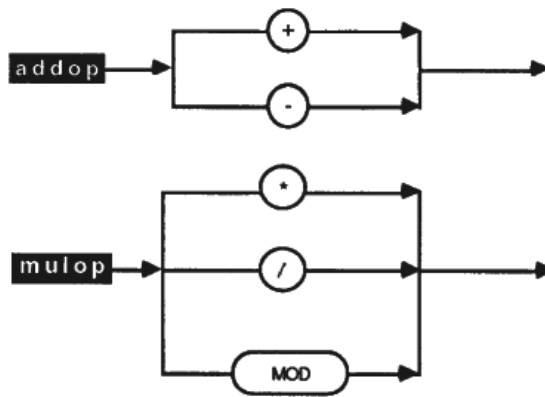
Priority	Operator	Meaning
3	**	Exponentiation
2	*	Multiplication
	/	Division
	MOD	Remainder
	NOT	Logical complement
1	+	Sum
	-	Difference, negation
	AND	Logical AND
	OR	Logical OR
	XOR	Logical exclusive OR

For example, in the expression $a + b * c$, $b * c$ is evaluated first because the $*$ operator has a higher priority. Operators with the same priority are evaluated left to right.

Arithmetic operators definition

The arithmetic operators are $+$, $-$, $*$, $/$, MOD and $**$; their usual meanings are as listed in “Logical Operators Truth Table” on page 40. Arithmetic operators can take only operands with data type NUMBER.

Arithmetic operators syntax



Description

The minus sign can be used to indicate subtraction (for example, $x-y$), or to indicate negation (for example, $-x$). As a negation operator, the minus sign cannot appear twice in a row.

The exponentiation, or power operator ($**$) works as follows:

If

$x > 0$	the result is as expected
$x = 0$	the result is always 1; the expected result would be 0, but the MFC's calculation uses the formula: $X ** Y = X^Y = e^{y \ln x}$, $X > 0$
$x < 0$	the result is $ X ** Y$

Exponentiation also produces a small difference in the calculated result if both X and Y are integers; for example $2 ** 10 = 2^{10} = 1023.9999$, instead of 1024.

Logical operators definition

The Logical operators are NOT, AND, OR, and XOR (exclusive OR), with their usual meanings, as shown in “Logical Operators Truth Table” on page 40. Logical operators can only take operands of type LOGICAL.

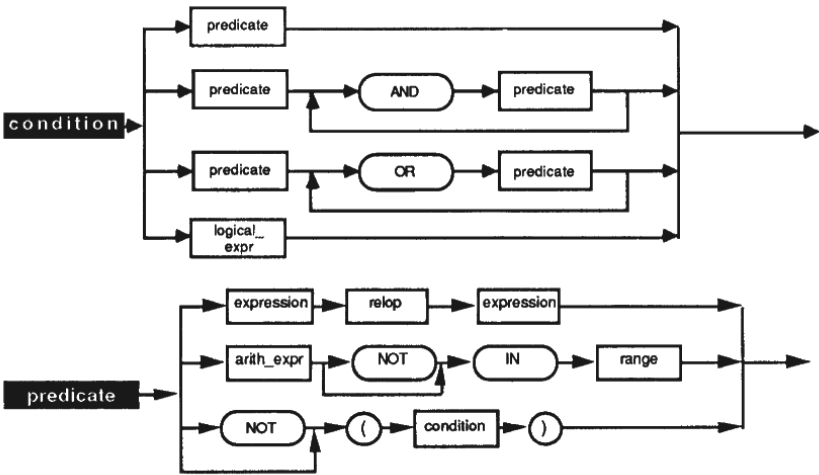
Table 21 Logical Operators Truth Table

a	b	NOTa	aANDb	aORb	aXORb
on	on	off	on	on	off
on	off	off	off	on	on
off	on	on	off	on	on
off	off	on	off	off	off

Conditions

A condition is a formula that expresses a truth or falsehood. It is an enhanced form of expression, used where a truth value is to be tested, as in an IF statement.

Conditions syntax



Description

A condition can be a logical expression, a relation between two expressions, a range test, two conditions joined by AND or OR, or any condition prefixed by NOT.

When a condition is a logical expression, it is implicitly tested for equality to ON. For example:

```
IF x AND y AND NOT z THEN ...
```

is equivalent to

```
IF (x = On) AND (y = On) AND NOT (z = On)
THEN ...
```

Relations definition

The relational operators are shown in “Relational Operators” on page 41.

Table 22 Relational Operators

Operator	Meaning
<	Less than
=	Equal to
>	Greater than
£	Less than or equal to
§	Greater than or equal to
◊	Not equal to

Not all relations are defined on every data type. The full set of six relations is defined only for expressions of type Number. The relations of equality and inequality are defined for discrete types. No relations are defined for arrays taken as a whole.

Range tests definition

Range tests (x IN y..z, a NOT IN b..c) are defined only on Number data type. The test is inclusive; for example, 5 IN 5..10 is true. The value of the left-most expression in range must be less than the value in the right-most expression.

Connecting conditions with OR and AND

The Logical operators AND and OR, but not XOR, can be used to connect conditions.

When both AND and OR are used in a compound condition, parentheses must be used to show grouping, just as when AND and OR are used as Logical operators.

For example:

$a < 5 \text{ AND } b = 6 \text{ OR } c > 7$

is ambiguous and must be rewritten as one of the following:

$(a < 5 \text{ AND } b = 6) \text{ OR } c > 7$

$a < 5 \text{ AND } (b = 6 \text{ OR } c > 7)$

Statements

This section describes the statements that you can use when building TDC CL structures. It contains the following major topics:

- Program statement definitions
- Embedded compiler directives

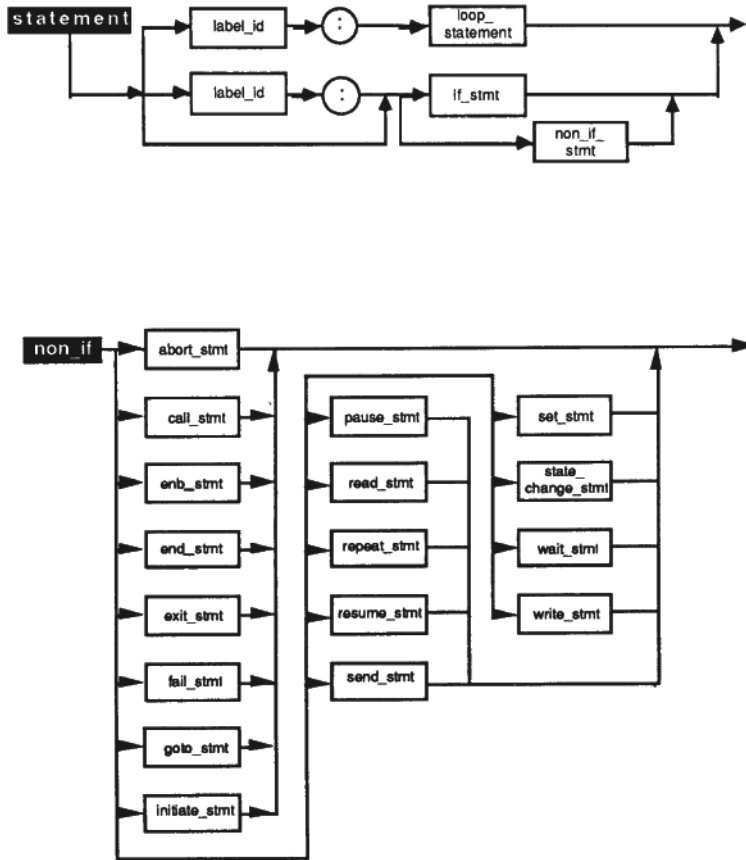
A statement defines a single, simple action to be performed within a TDC CL structure. TDC CL statements can be grouped into two major categories, according to function: Program statements and Embedded compiler directives.

TDC CL Program Statements can be Categorized as follows:

- Assignment statements
whose purpose is to change the value of one or more variables: SET, READ, WRITE, and the state-change statement.
- Control statements
which establish program conditions or direct the flow of control within a program: GOTO, IF/THEN/ELSE, LOOP/REPEAT, CALL, ENB, INITIATE, and RESUME.
- Delaying statements
which cause the program to wait for some event to occur or for a time delay: PAUSE and WAIT.
- Termination statements
which signify the termination of the program or of a part of it: FAIL, EXIT, ABORT, and END.
- Communication statements
which communicate with an operator: SEND.

Program statement definitions

Program statement syntax



Statement labels

Labels are used as the targets of GOTO and REPEAT statements. GOTO and REPEAT are used to transfer control to another part of a program, which is identified by the label referred to in the GOTO and REPEAT statements. A label is an identifier followed by a colon. A label can precede any statement, but cannot appear on a continuation line; therefore, a statement that is embedded within another statement cannot have a label. A LOOP statement must have a label; therefore, a LOOP statement cannot be embedded within another statement.

Example

```

lab_01:      CALL test (x, y, z)
IF x > y THEN (SET x= z;
& badlabel : SET z = y) - NOT VALID

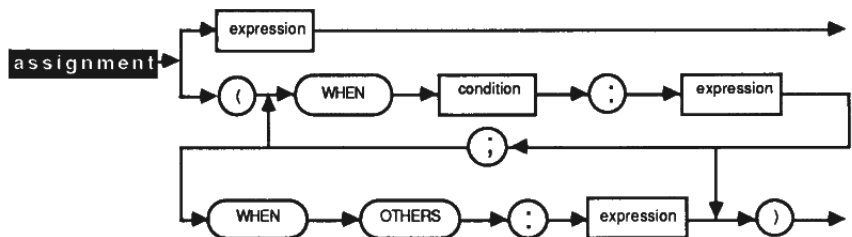
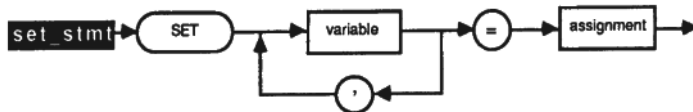
```

In this example, badlabel appears on a continuation line; therefore, it is invalid.

SET statement

This statement modifies the value of one or more variables, possibly depending on the result of one or more conditions.

Syntax



Description

A SET statement with a simple expression on the right-hand side of the equal sign unconditionally assigns the value of that expression to each of the variables on the left-hand side of the equal sign. The assignments are executed in reverse order from the order in which the variables are declared.

A SET statement whose right-hand side begins with (WHEN... is called a conditional SET statement. When this statement is executed, its WHEN conditions are successively evaluated, until a true condition is found. The expression corresponding to the true condition is then evaluated and its value is assigned to the variables on the left-hand side of the equal sign. After one true

condition is found, no other conditions are evaluated. Execution proceeds with the next statement. The data type of the assignment on the right hand side must match the data type of the variable(s) on the left-hand side.

A conditional SET statement can have any number of WHEN clauses. The last WHEN clause can name the special condition OTHERS, which is always true.

A conditional SET statement that does not name WHEN OTHERS can fail; that is, none of the conditions may be true. If this occurs, it is a run-time error. A maximum of 16 parameters can be referenced with one SET statement.

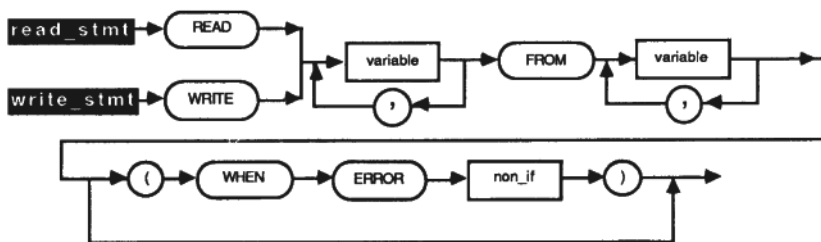
Example

```
SET x = x + 1—simple SET
SET x, y, z = 0—multiple SET
    - first z is set to 0 then y is set to 0
    - then x is set to zero last
SET color = (WHEN x > lim: red;
    - conditional SET
& WHEN x <lim-db: green;
& WHEN OTHERS: yellow)
IF A100.MODE <>PMAN
    - testing mode before
    - change avoids
& THEN SET A100.MODE = PMAN
    - unnecessary set of
    - slot change flag
```

READ/ WRITE statements

These statements are used to access remote variables in other MFCs on the same C-Link and test to see whether the access was correctly performed.

Syntax



Description

The READ statement reads local variables (into this MFC) from remote variables (in other MFCs). The WRITE statement writes remote variables (into other MFCs) from local variables. The number of variables on the left-hand and right-hand sides of the FROM must be equal.

The number of variables allowed in a READ statement is limited to 16. The number of variables allowed in a WRITE statement depends on the type of the variables: numeric variables, 15; flag variables 16; combined numerics and flags in a single WRITE 16 with no more than 14 numerics.

The program is suspended until the system confirms transmission of all variables. If all are correctly transmitted, the program proceeds to the next sequential statement. If there is any communication error - so that one or more variable cannot be correctly transferred - the statement in the error clause is executed. If there is no error clause and a communication error prevents a variable transmission, the calling sequence program is failed.

These statements are Preemption points. A READ or WRITE statement without an error clause differs from the corresponding SET statement only in that it is a preemption point.

If variables in the other MFCs are accessed by their tag names, these tag names must be listed in the EXTERNAL declarations. If the variables are accessed by their system IDs, the box Data point ID (that is, \$НУХХВУУ) must be included in the EXTERNAL declarations. You should read all of the sections in “External data point parameters” on page 76 to understand methods of parameter access with TDC CL.

Only the parameters, of variables in other MFCs (shown in the following table) can be accessed through the READ/WRITE statements. In the table, R indicates that read access is allowed, W indicates that write access is allowed, X indicates that no access is allowed, and - means that the Point has no such parameter.

Table 23 Parameters Accessible Through the C-link

Type of Point	OP	PV	PVP	STATE	SP	SPP
Analog input	-	R	R	X	X	X
Analog output	R	-	-	-	-	-
Counter input	-	X	-	R	-	-
Digital input	-	R	-	-	-	-
Digital output	R	X	-	-	-	-
Flag	-	R/W	-	-	-	-
Logic block	-	R	-	-	-	-
Numeric	-	R/W	-	-	-	-
Regulatory ctrl	X	R	R	-	R	R

Table 23 Parameters Accessible Through the C-link (*continued*)

Type of Point	OP	PV	PVP	STATE	SP	SPP
Timer	—	X	—	R	X	-

Example

```

LOCALa AT NN (01)
LOCALb AT NN (02)
LOCALc AT NN (03)
LOCALd AT NN (04)
LOCALE AT NN (05)
EXTERNAL AIOO, BIOO, timer1
LOCAL temp1 : OPEN/CLOSE AT f1(01)
LOCAL temp 2 : STATE AT f1(02)
...
READ a, b FROM A100.PV, B100.PV
    (WHEN ERROR GOTO foo)
READ c, d FROM A100.SP, B100.SP
READ temp2 FROM timer1.state
    - reading a timer state into a
    - local variable
WRITE B100.PV FROM e (WHEN ERROR FAIL)

```

Communication error handling

A communication error on a WRITE, or any other store, indicates that one or more of the local variables is not actually stored into its corresponding remote variable.

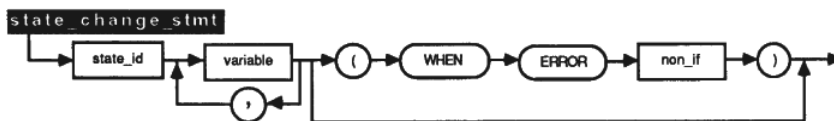
In the MFC, if a communication error occurs during a READ, one or more destination variables remain unchanged.

In event of a communications error, the WHEN ERROR clause can be used to start at an appropriate action; if there is no error clause, the calling sequence is failed.

STATE CHANGE statement

This statement sets the state of the Output (OP) parameter of one or more Digital output Data points that have two or more discrete output states and optionally verifies that the state has been properly set.

Syntax



Description

The variables must identify Data points that have an OP parameter of a discrete type. The STATE ID (which is the descriptor you used when you built the Point) can be of any discrete type, including Logical, but if more than one Data point is named, each OP must be of the same type.

This statement sets the Data points' OP parameters equal to the named state; therefore, close A100 is the same as SET A100.OP = close.

If the point identifier is omitted, the bound Data point is implied. A maximum of 16 parameters can be referenced with one state-change statement.

STATE CHANGE with feedback

The WHEN ERROR clause applies to only digital composite points, and when Command Disagree alarming is enabled. For other Point types and when Command Disagree alarming is not enabled, the WHEN ERROR clause is ignored.

The WHEN ERROR clause is executed only on feedback error; for other types of state change command failures the sequence is failed. The WHEN ERROR clause is executed if any variable in the variable list has a feedback error.

The following table describes the sequence program action on various conditions of the STATE CHANGE statement.

A STATE CHANGE statement with an error clause is a Preemption point.

	Command Disagree is Configured		Command Disagree is NOT Configured	
	WHEN ERROR path is coded	WHEN ERROR path not coded	WHEN ERROR path is coded	WHEN ERROR path not coded
Normal Execution	Seq. continues to next statement	Seq. continues to next statement	Seq. continues to next statement	Seq. continues to next statement
Feedback Timeout	WHEN ERROR path is executed	Seq. continues to next statement	Not applicable	Not applicable
Command Failure	Sequence fails	Sequence fails	Sequence fails	Sequence fails

Example Suppose

Motor1

valve2, valve3

37SW

output states are start/stop

Output states are open/close

output states are low/high

Then:

stop MOTOR1

open valve2, valve3

high 37SW (WHEN ERROR GOTO STEP retry)

are valid STATE CHANGE statements.

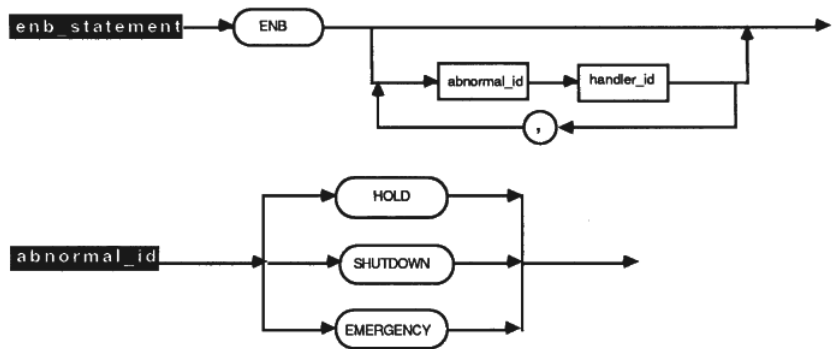
ENB statement

This statement enables a new set of abnormal condition handlers, or disables all currently enabled abnormal condition handlers.

When this statement is executed, all abnormal condition handlers that are currently enabled are disabled and only the specified handlers are enabled. The same handler type (for example, HOLD handler) cannot appear twice in the same ENB statement. If no handler names are specified, all handlers are disabled.

This statement is not a Preemption point and the new handler list does not take effect until the next Preemption point.

Syntax



Example

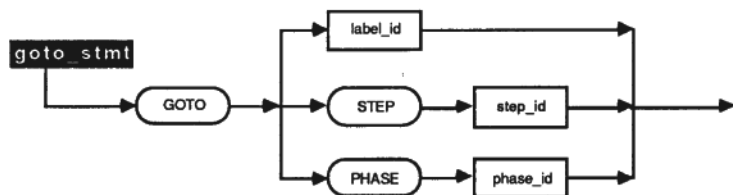
```

ENB HOLD fillhold, EMERGENCY ovenstop
ENB SHUTDOWN tankck
IF NN(3) > 55.0 THEN ENB HOLD hold1,
    EMERGENCY emer2
ELSE ENB HOLD hold2, EMERGENCY emer2
ENB
  
```

GOTO statement

This statement unconditionally branches to another place in the program.

Syntax



Description

In a sequence program, the target of a GOTO statement can be any label in the present step, the heading of any step in the current phase, or the heading of any phase in the program.

A GOTO statement cannot be used to exit a subroutine. Use EXIT instead.



Note

In the MFC, a backward-branching GOTO causes preemption. Although a forward GOTO statement is not a Preemption point, a GOTO STEP or GOTO PHASE statement always causes preemption, because STEP and PHASE headings are Preemption points.

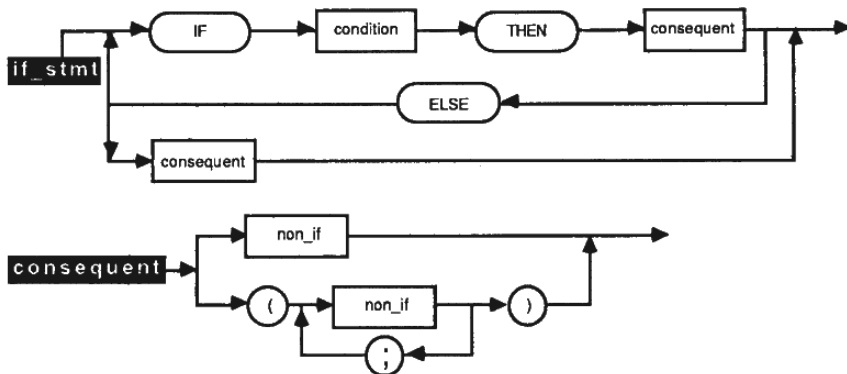
Example

```
GOTO label
  - simple GOTO
. . .
PHASE test
  - each execution of 'GOTO PHASE test' causes a phase
  - change to be placed in the MFC's Sequence
. . .
GOTO PHASE test
  - Circular List
PHASE test
  - this brings you back to beginning
label:
  - of the code for this phase without adding to the list
GOTO label
```

IF, THEN, and ELSE statements

These statements cause the conditional execution of another statement or statements.

Syntax



Description

Any ELSE or ELSE IF statement that does not directly follow an IF or ELSE IF statement is an error.

The consequent gives the statement(s) to be conditionally executed. The first form of the consequent indicates the conditional execution of a single statement; the second form indicates conditional execution of multiple statements. Consequents are considered one statement for syntax checking; therefore, if consequents are on a separate line from the IF, a continuation character is required for each line.

A sequence of IF ... ELSE IF statements is evaluated until one of the IF conditions is true. If this occurs, the consequent of the statement that has the true condition is executed. Any succeeding ELSE IF or ELSE statements in the sequence are ignored.

If none of the conditions in the IF and ELSE IF statements are true, the consequent of the ELSE statement (if any) is executed.

An IF, ELSE IF, or ELSE statement is not a Preemption point of itself; however, the consequent of an IF, ELSE IF, or ELSE can contain one or more Preemption points.

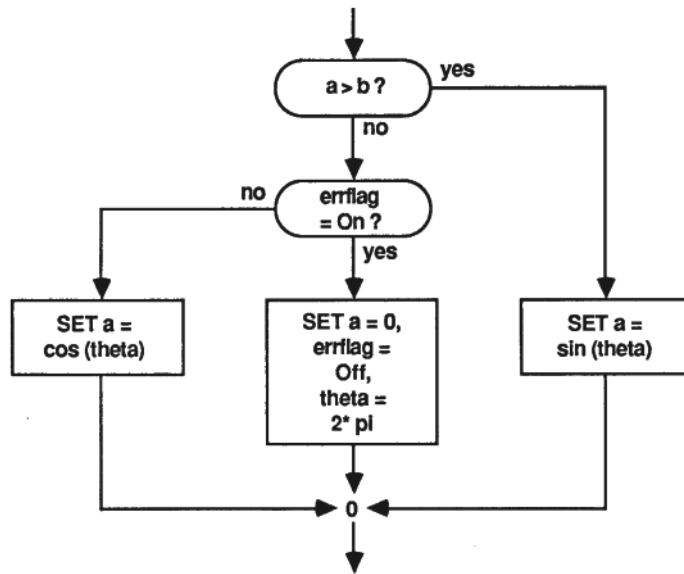
An IF or ELSE statement must always appear on a new line (you need an & for a THEN... that appears on a new line but not for ELSE...). It can be indented like any non_IF statement. It can never appear in the consequent of an IF or ELSE statement, in a WAIT statement's WHEN clause, or in the error clause of a READ, WRITE, state change or INITIATE statement.

Example

```
IF 2b OR NOT 2b <>the_question THEN FAIL
IF x < y THEN SET x = y
IF x NOT IN 1..10 THEN
    (SEND: "range error", x;
    &                GOTO retry)
IF a > b THEN SET a = sin (theta)
ELSE IF errflag THEN (SET a = 0;
    &                SET errflag = Off;
    &                SET theta = 2 * pi)
ELSE SET a = cos (theta)
```

The second example above corresponds to the flowchart in "IF Statement Flowchart" on page 54.

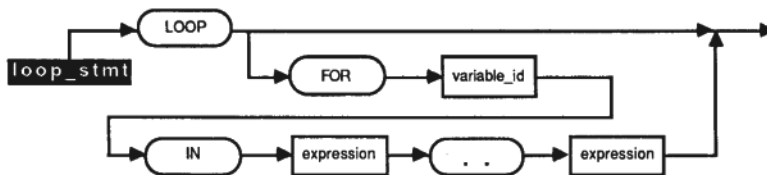
Figure 7 IF Statement Flowchart



LOOP statement

This statement provides loop control in a step or subroutine.

Syntax



Description

The LOOP statement's FOR clause names a counter variable. This variable must be a scalar local variable or scalar subroutine argument of type Number. The upper and lower bounds of the range are evaluated. If their values are not integers, they are rounded to the nearest integer.

The counter variable is initialized to the value of the lower bound. Each time that loop's REPEAT statement is executed, the counter variable is incremented by 1. If that value does not exceed the range's upper bound (previously computed), the loop is repeated; otherwise the loop terminates. On normal exit from the loop, the counter variable is equal to the final expression plus 1.

On the MFC, the upper bound of the FOR range is dynamically re-evaluated each time the REPEAT statement is executed. If the LOOP statement does not contain a FOR clause, it never normally terminates. It can be exited by a GOTO or EXIT statement, or by the occurrence of an abnormal condition. Each backward branch on the REPEAT statement causes preemption.

A LOOP statement must have a label.

Example	<pre>label :LOOP label :LOOP FOR count IN 10..20</pre>

REPEAT statement

This statement causes a loop to be repeated.

Syntax



REPEAT description

The target of a REPEAT statement must be a label in the current step, block, or subroutine.

The label ID must define a loop (that is, the label referred to must have a LOOP statement attached to it).

A loop can have only one REPEAT statement. REPEAT statements are contextually bound to the LOOP statement having the given label.

The REPEAT statement causes the loop's counter variable (if any) to be incremented by 1. If the counter variable is then less than or equal to its final value, the program branches back to the first statement in the loop. If the counter variable exceeds its final value, the branch is not taken and execution proceeds sequentially, following the REPEAT statement.

If the loop does not define a counter variable, the REPEAT statement causes an unconditional branch to the first statement in the loop.

Loops can be nested to any depth. Whenever a loop is entered through its heading (or its beginning), its loop counter is reset, and it again begins counting towards its maximum.



Note

A REPEAT statement is a Pre-emption point in the MFC.

Example

```
setx:LOOP FOR i IN 1..5
    SET x.SP = 2
    WAIT 30 SECS
REPEAT setx
```

The following example demonstrates nested loops:

```
outer :LOOP FOR i IN 1..10
inner:LOOP FOR j IN 1..i
    SET a (i, j) = -1.0
    REPEAT inner
REPEAT outer
```

PAUSE statement

If a sequence program is in semiautomatic mode, this statement causes it to pause until it is resumed by the operator. In fully automatic mode, the PAUSE statement is ignored.

This statement is a Preemption point.

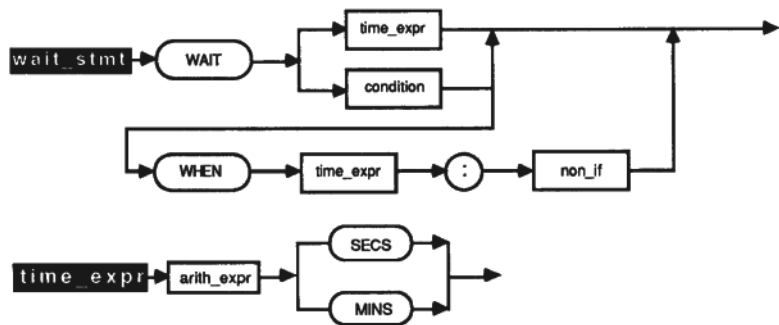
Syntax



WAIT statement

This statement causes the program to wait until some condition is fulfilled, or until a timeout occurs. This statement is a Preemption point.

Syntax



Description

The “WAIT time_expr” form of the WAIT statement defines a simple timed delay.

The time expression cannot contain mixed units; it can be MINS or SECS but not both. Because of asynchronous MFC timekeeping, the delay can terminate as early as one time unit before the specified time.

For the “WAIT condition” form of the WAIT statement, the program is suspended until the named condition becomes true. If the WAIT statement contains a WHEN clause, the WAIT is timed out as specified. If the time expires while the condition is still false, the statement in the WHEN clause is executed.

Example

```

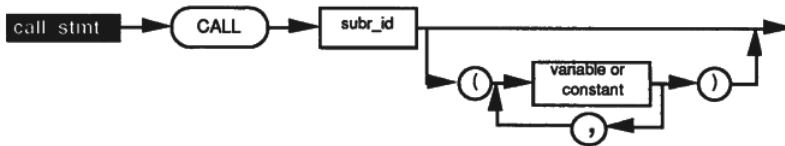
WAIT 5 MINS
WAIT x >5.5
WAIT A_001.PV IN A_001.lolim .. A_001.hilim
  boil:LOOP FOR tick IN 1..60 - minutes
    IF temp >= 100 THEN GOTO done
    WAIT press > 1(WHEN 1 MINS: GOTO next)
    ...
    - do something to adjust pressure
  next: REPEAT boil
    - go test temp again
    SEND: "this watched pot never boiled"
    FAIL
  done:...
    - ok, continue with process
  
```

In the last example, note that temp is tested once every minute, as long as press is less than 1; but press is continuously tested.

CALL statement

This statement invokes (calls) a subroutine.

Syntax



Description

The arguments of the CALL statement are limited to variables or constants that must match the arguments of the subroutine declaration in both data type and mode (IN, OUT, or IN OUT). A variable must appear in each place where the subroutine heading names an OUT or IN OUT argument; a variable or constant can be used in the place of an IN argument.

In TDC CL, this statement cannot occur within a handler or subroutine.

Example

```

EXTERNAL HG050601-   HG   digital input point
EXTERNAL HG0502-     HG   regulatory point
EXTERNAL $HY02B05-   HG   BOX data point
LOCAL num AT NN(3)
LOCAL flagarr : Logical ARRAY (1..3) AT
                  FL(10)
...
CALL sub1 (HG050601.PV, DI(116).PV
           - digital input PVs
CALL sub2 (num, flagarr)
           - scalar local variable, entire
           - array local variable
CALL sub3 (flagarr(2), 5.5, $HY02B05.NN(19)
           - element of local array, numeric
           - literal, array parameter indexed
           - by a constant
CALL sub4 (HG0502.SP)
           - scalar parameter
CALL sub5 (flagarr(num))
           - ILLEGAL the local array

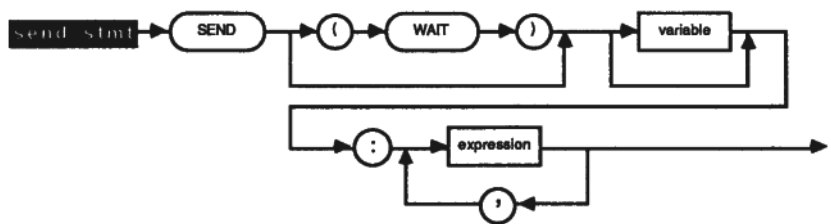
```

- is not indexed by a constant
- ```
CALL sub6 (state1)
```
- enumeration state identifier
- 

## SEND statement

This statement sends a message to the operator or to another Data point. It can optionally wait for operator confirmation.

## Syntax



## Description

The optional “variable” to the left of the colon establishes the message destination as follows:

|          |                              |
|----------|------------------------------|
| none     | to home unit MMI CRT and LOG |
| CRT_Only | to home unit MMI CRT only    |
| LOG_Only | to home unit MMI LOG only    |

The WAIT option suspends the program until the message is confirmed by the operator. If WAIT is specified the SEND statement is a preemption point. The WAIT option can be selected with only (none) and CRT\_Only.

The “expressions” in the SEND statement are the message items to be sent. Each message item can have a value of any of the types: Number, Logical, String literal, or 2-state enumeration. An array cannot be sent as a whole, but its elements can be individually sent. Tag names cannot be sent in the SEND statement.

SENDing only a null string or string of all blanks is not allowed; a SEND that sends other items in addition to a blanks string is allowed. Each word in a SEND string literal must be eight characters or less.

For Destinations (none) CRT\_Only and Log\_Only, there can be no more than seven Send items, where a Logical counts as one item, a Number counts as two items, and a String counts as one item. Messages with seven Strings sent to CRT have the last two characters of the last String truncated. The message correctly appears at the LOG.

---

|                |                                                                                                                                                                                                                 |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Example</b> | <pre> SEND: "Send examples" SEND (WAIT) CRT Only: "Value is", LP(I).PV SEND (WAIT): "Error ", A100.PV, "out of range" SEND Log_Only: "values are", a, b, "and", c SEND CRT_Only: "This is a CRT message" </pre> |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

---

### Preemption of SEND statement

When an abnormal condition occurs and a handler is entered, the main sequence might have been waiting for operator response on an outstanding SEND statement with the WAIT option. Any such statements are aborted when the handler begins to execute.

This is done for two reasons. First, the handler may need to execute a SEND with WAIT. Second, it is likely that when the handler is done, the program will no longer need the confirmation that it was waiting for.

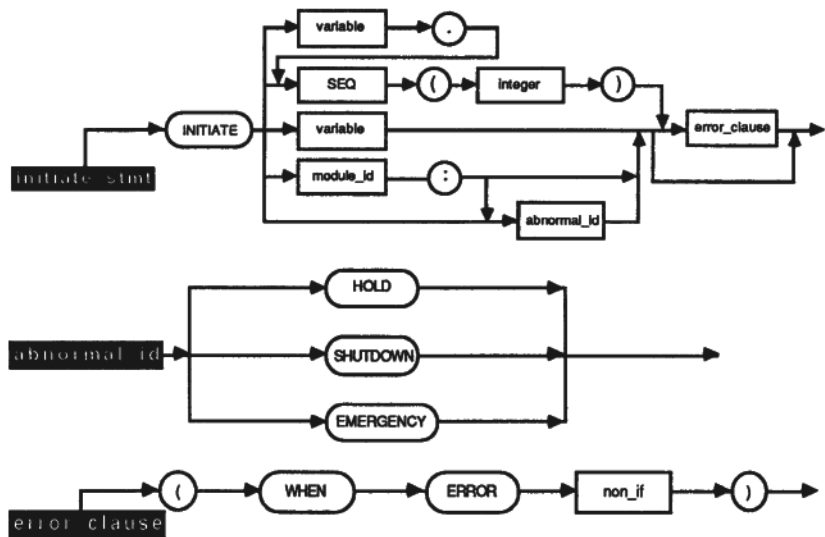
### INITIATE statement

This statement causes the initiation of an abnormal sequence (that is, an Abnormal Condition Handler) of the sequence program executing the INITIATE statement; or it can start a separate sequence program or an abnormal sequence of a separate sequence program.

It can optionally wait for confirmation that the requested action has occurred and take action on communication errors when initiating a process module in another MFC.



## Syntax



### Description: initiating data points and programs

**INITIATE variable** starts another sequence in this MFC; the variable must be a Process module Data point identifier, and must be declared in an **EXTERNAL** statement.

**INITIATE SEQ(n)** where n is an integer corresponding to the Sequence slot number, starts a sequence in this MFC.

**INITIATE variable.SEQ(n)** starts a sequence in another MFC; the variable must be an MFC box Data point identifier (of the form \$HYnnBmm, see “Box data point identifiers” on page 22) and must be declared in an **EXTERNAL** statement.

**INITIATE module\_id** where the module\_id must name a Process module Data point, starts the sequence program that is bound to that Point.

### INITIATE Description: initiating abnormal condition handlers

**INITIATE module\_id:abnormal\_id** starts the named Abnormal Condition Handler of the sequence program executing in the named module, as soon as the sequence reaches its next Preemption point.

**INITIATE abnormal\_id** starts the named handler of the present sequence program. That handler must be higher in priority than the one executing this

statement. Control is directly transferred to the head of the handler, no further statements are executed by the present sequence program.

An INITIATE statement that starts an Abnormal Condition Handler of the same sequence program, while not a Preemption point itself, transfers control to a preemption point, thus causing preemption.

### INITIATE Description: WHEN ERROR clause

The INITIATE request can result in an error condition for several reasons, including, the target Process module is in the wrong state; the target process module is not loaded; the requested handler is not enabled; or there is a communications failure between MFCs. In most instances, in case of such error, the requesting sequence is failed.

The WHEN ERROR clause is activated only when a communication error between MFCs occurs (the statement in the error clause is executed and the requesting sequence is not failed).

It is a compile-time error to use a WHEN ERROR clause on an INITIATE of your own abnormal condition handler. The compiler will accept a WHEN ERROR clause on an INITIATE of another Process module in the same MFC, but it will have no effect.

An INITIATE statement with a WHEN ERROR clause is a Preemption point.

---

#### Example

```
EXTERNAL mixer, reactor
 - Process module data points
EXTERNAL boiler, oven
 - Process module data points
EXTERNAL $HY02B10
 - Box data point identifier of another MFC
INITIATE mixer
 - Starts another sequence in this MFC
INITIATE mixer:hold
 - Starts handler of another sequence in this MFC
INITIATE EMERGENCY
 - Starts handler in this sequence in this MFC
INITIATE $HY02B10.SEQ(10) (WHEN ERROR GOTO retry)
 - Starts sequence in sequenceslot 10 in
 - another MFC with goto action when error
INITIATE $HY02B10.SEQ(6):EMERGENCY
 (WHEN ERROR INITIATE SHUTDOWN)
 - Starts handler of program in sequence slot 6
 - in another MFC with handler of this sequence
 - to start when error
INITIATE SEQ(2)
 - Starts sequence in slot 2 in this MFC
```

---

**FAIL statement**

This statement causes the program to be suspended and enter a soft failure state. The operator is informed and can take recovery action. The recovery actions available are system-defined; they include, but are not limited to, resuming execution of the program at the next sequential statement (presumably after having made some changes).

This statement is a Preemption point.

**Syntax****Example**

```
close switch_2 (WHEN ERROR FAIL)
IF val > 10 THEN (FAIL; GOTO recover)
```

**RESUME statement**

This statement can be executed by only the Restart routine of an Abnormal Condition Handler. It causes resumption of the normal sequence at the beginning of a specified phase (which allows preemption to occur).

**Syntax****Description**

The named phase is resumed at its head.

**EXIT statement**

When this statement is executed in a subroutine, it returns control to the subroutine's caller. When executed in a main program or Abnormal Condition Handler, it terminates the program.

**Syntax****Description**

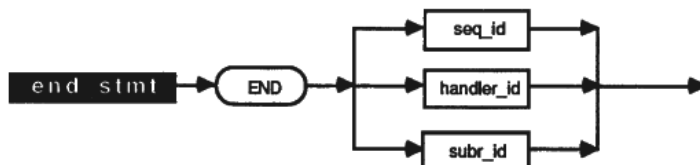
A program termination resulting from the execution of an EXIT statement is considered a normal termination (as opposed to that caused by the ABORT statement). Executing an EXIT statement is equivalent to falling off the end of a program.

**ABORT statement**

This statement causes program termination. When executed in a subroutine, it terminates both the subroutine and its caller.

**Syntax****END statement**

This must be the last statement of a main program, subroutine or handler. Execution of this statement is identical to the EXIT statement.

**Syntax**

**Description**

The END statement is counted as an executable statement even though it is not assigned a statement number on program listings.

The ID in each END statement must match the ID in the corresponding sequence or handler or subroutine heading.

## Embedded compiler directives

This section describes Compiler directives that can be directly embedded in a CL/MFC structure.

### Syntax

All Compiler directives begin with a percent sign (%) and must begin in the first column of a source line. Alphabetic characters can either be upper- or lower-case.

### %PAGE directive

This directive has no effect in the current version of the TDC CL Compiler.

### %DEBUG directive

This compiler directive effects conditional compilation. There is a global compiler variable called the compiler debug switch that can be turned on or off when you compile your structure, but cannot be changed during a compilation.

When the compiler debug switch is off, any source line that begins with %DEBUG is ignored.

When the compiler debug switch is on, each source line that begins with %DEBUG is treated as an ordinary source line with the %DEBUG stripped off.

---

#### Example

```
set x.PV = 4
%DEBUG SEND: "x.PV has been sent"
```

In this example the SEND statement is executed only if the compiler debug switch is on at the time the program is compiled.

---

---

# Structures

This sections has information specifically about TDC CL for the Multifunction Controller. It contains the following major sub-topics:

- Sequence program definition
- Abnormal condition handlers
- Restart routines
- Accessing parameters
- User-written subroutines
- Built-in functions and subroutines

TDC CL structures can be independently compiled and therefore are referred to as compilation units. They are composed of a series of headings, data definitions, and/or statements that execute a custom control strategy. The TDC CL structure (sequence program) executes on only the Multifunction Controller or Advanced Multifunction Controller).

The smallest executable instruction in a TDC CL structure is the statement. A program statement is a command to perform a simple action, and is used in TDC CL sequence programs, subroutines, and Abnormal condition handlers. Some program statements have the side effect of preempting (suspending) program execution. All Program statements are described in “Statements” on page 43.

## Sequence program definition

### General

A sequence program is a set of instructions that details a complete sequence of events in the production of some product. A sequence program is loaded (after compiling without errors) into a Process module, which is a named data point in the MFC. Sequence programs are loaded from the Process module detail display.

Sequence programs are constructed by combining statements to form steps, combining steps to form phases, then combining phases to form the sequence of tasks you want to perform. Subroutines can exist within sequence programs. The aspects of subroutines that apply to TDC CL are discussed in “User-written subroutines” on page 87 and “Built-in functions and subroutines” on page 89.

Abnormal condition handlers are TDC CL structures that can be used to take control from the sequence program if a certain “abnormal condition” occurs, usually some sort of process upset.

A sequence program’s execution can be interleaved with that of other sequence programs running in the same Multifunction Controller.

The suspension of a sequence program to allow another program to run is called preemption.

### Sequence program description

A sequence program consists of a main sequence, optionally followed by Abnormal condition handlers and subroutines.

The main sequence is made up of phases; a phase or subroutine consists of one routine. An Abnormal condition handler consists of one or two routines, the second (if present) being its Restart routine. Routines (phases, handlers, and subroutines) are made up of steps; however, if a routine consists of only one step, the STEP heading may be omitted.

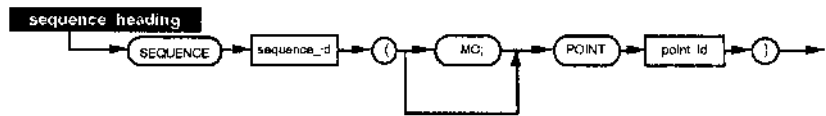
The sequence ID (handler ID, subroutine ID) in each END statement must match that in its respective heading.

### SEQUENCE heading

The SEQUENCE heading identifies the Sequence program and the bound Data point.



## Syntax



## Description

The sequence ID is an identifier by which the program is externally known. The Point description identifies the bound Data point, which must be a process module Data point.

### Example

```

SEQUENCE fred (POINT PM47B)
or
SEQUENCE fred (MFC; POINT PM47B)

```

## PHASE heading

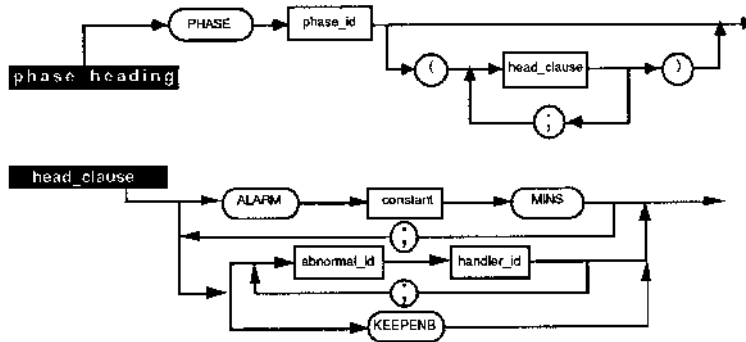
A phase is a major process milestone. Phase boundaries are key synchronization points in the sequence program. The PHASE heading identifies the beginning of a new phase and sets up the operating conditions for the execution of its routines.

During a phase, a check is made at each Preemption point for the occurrence of abnormal conditions. When such a condition occurs, all activities cease and an Abnormal condition handler is activated.

A phase-alarm timer, if active, checks the total execution time of a phase.

At the end of a phase, all activities, including abnormal condition detection, cease. The phase alarm timer is stopped. The sequence program is momentarily quiescent, and a new phase may begin. Each time a phase boundary is crossed, a phase change is noted in the MFC's Sequence circular list. In the new phase heading, the phase-alarm timer can be restarted at a newly specified setting and new Abnormal condition handlers can be activated.

## Syntax



## Description

The **ALARM** heading clause sets the phase alarm timer to the value of the time expression. If this clause is omitted, the phase-alarm timer is not started for this phase. The alarm constant must be an integer in the range of 1 to 9999.

The **HOLD** heading, **SHUTDOWN** heading, and **EMERGENCY** heading clauses enable handlers for their respective abnormal conditions. Activation conditions for these handlers are defined in the handlers' own headings. The same Abnormal condition handler heading clause cannot appear twice in one **PHASE** heading. If a **PHASE** heading contains no clause that activates a given handler, that condition is disabled.

The **KEEPENB** clause specifies that the current set of enabled handlers should not be disabled and should be retained. A **PHASE** heading can contain either the **KEEPENB** clause or named handlers, but not both.

Note that based on program flow, you may not be sure which handlers are enabled when a **PHASE** heading is encountered. If you need to be certain, name the required handlers in the **PHASE** heading. If no handlers and no **KEEPENB** clause are specified in the **PHASE** heading, all handlers are disabled. A compiler note is issued for this condition.

### Example

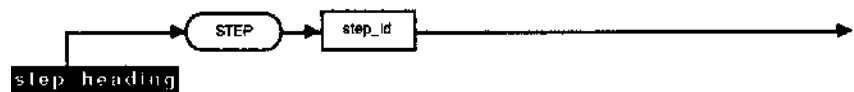
```
PHASE fill_up (ALARM 2 MINS;
& HOLD fillhold)
PHASE baking (ALARM 30 MINS;
& SHUTDOWN oven_clr
& EMERGENCY oven_stp)
```

## STEP heading

A step is a named minor milestone of the process, that consists of one or more statement groups separated by a STEP heading. As a process milestone, a step is recognized and displayed at the Universal Station.

The STEP heading names a step.

## Syntax



## Description

A step change is marked for the sequence.

The STEP heading is a Preemption point.

---

### Example

```
STEP s1
STEP FILL_A
```

---

## Abnormal condition handlers

When a sequence program is not handling an abnormal condition, it is said to be in its normal sequence; otherwise, it is in an abnormal sequence. See “Statements” on page 43 for complete information on any of the statements mentioned in this discussion (for example, SEND).

The Abnormal condition identifiers are HOLD, SHUTDOWN, and EMERGENCY. These are reserved words.

Abnormal condition handlers have priority over each other and over normal sequences. These priorities are defined as follows:

```
EMERGENCYHighest
SHUTDOWN.
HOLD.
Normal sequenceLowest
```

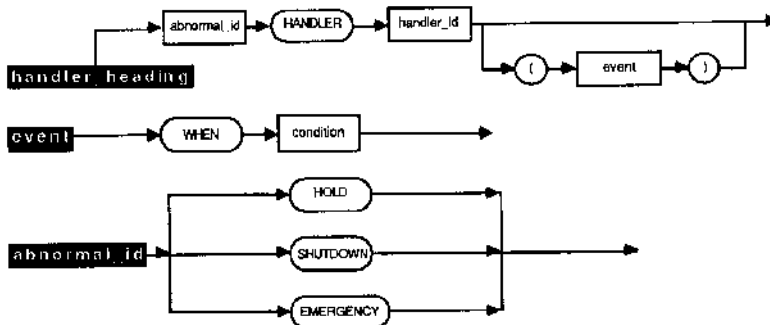
If the operator, or any INITIATE statement (in the same or another program), should attempt to start an Abnormal condition handler that is not enabled, the condition is Propagated to the next lower priority. That is, if an attempt is made to start EMERGENCY and there is no enabled EMERGENCY handler, the SHUTDOWN handler is started. If that handler in turn is disabled, the HOLD handler is started.

An attempt to start a HOLD handler that is not enabled causes the program to enter the FAIL state, as if a FAIL statement had been executed.

### HANDLER heading

The HANDLER heading specifies the name of the handler, the Abnormal condition it handles, and its starting condition (if any).

### Syntax



## HANDLER heading description

The handler ID is the identifier by which the handler is to be known. The abnormal ID (HOLD, SHUTDOWN, or EMERGENCY) defines the condition to be handled.

An Abnormal condition handler must be enabled by a PHASE heading. All handlers are disabled at the end of each phase. A handler cannot start until it has been enabled.

Once enabled, a handler can be started in one of three ways:

- by operator action
- by execution of an INITIATE statement in any sequence program (including its own)
- by the occurrence of the event (if any) named in its heading

The WHEN condition is tested ONLY at each Preemption point. A PHASE header first enables handlers and then checks for the condition in the WHEN clause; therefore, you must have a Preemption point between the condition (that you are checking for in the WHEN clause) and the next PHASE header. Then, if it is true, the WHEN event is deemed to have occurred and the handler begins execution. The following example shows a Preemption point STEP S1, which is between the condition being set to true (SET x = On) and the next PHASE header (PHASE p3).

### Example

```
SEQUENCE s (POINT p)
LOCALx: logical AT FL(1)
PHASE p1
 SET x = Off
PHASE p2 (HOLD h) STEP S0
 SET x = On
STEP S1
 GOTO PHASE p3
PHASE p3
 SET x = Off
END s
HOLD HANDLER h (WHEN x = On)
 SEND: "hold handler beginning"
RESTART
 RESUME PHASE p3
END h
```

A handler without a given event can be started by only operator or program action.

Once in an Abnormal condition handler, the run time behaves as if that handler and all lower priority handlers were disabled. This is because the condition that triggered the handler may still be true which, if the current handler were still

enabled, would re-trigger the handler again, causing a loop. (Lower priority handlers are not called while in a higher priority handler.)

When you exit the Abnormal condition handler (with the RESUME statement), that handler and lower priority handlers remain disabled. If you want to re-enable that priority handler or lower priority handlers, an ENB statement is required.

---

**Example**

```

SEQUENCE seq1 (POINT p)
 PHASE one (SHUTDOWN sh1;EMERGENCY em1)
 STEP sone
 SET NN(3) = NN(3) - 1
 IF NN(3) >10.5 THEN SET FL(1) = Off
 STEP stwo
 - FL(1) is off, so sh1 is entered
 ...
 PHASE two (KEEPEPB)

 SHUTDOWN HANDLER sh1 (WHEN FL(1) = Off)
 STEP shone
 - at this point the runtime "disables" sh1, but
 - EMERGENCY handler em1 is still enabled
 - shutdown corrective action is done here

 RESTART
 - preparation to return to
 - the main sequence is done here

 RESUME PHASE two
 - sh1 is not re-enabled on
 - return to the main sequence
 EMERGENCY HANDLER em1(WHEN FL(2) = On)
 HOLD HANDLER cooldown
 (WHEN Temp.PV Temp.PVHITP)

```

---

## Restart routines

HOLD and SHUTDOWN Condition handlers can be terminated by a RESTART routine, separated by a RESTART heading. The RESTART routine is intended to contain statements that prepare for re-entry into the normal sequence. The optional Resume statement specifies the phase label where normal execution will begin.

A Restart routine has the same priority as the Abnormal condition handler to which it is attached. The Module operating status on the Process module detail display does not change as the Restart routine is entered. In addition, because the restart section is not functionally separate from its parent handler, the detail display does not show that a restart handler has been enabled when the sequence is executing the restart section; therefore, you should inform the operator that the program is executing a restart, using either a descriptive STEP label in the restart section, or SENDING a message.

The RESUME statement is optional and can appear anywhere inside a RESTART routine. If you do not include a RESUME statement, execution of the sequence terminates at the end of the RESTART routine.

No Operator intervention is required to confirm execution of a RESTART routine or a RESUME statement.

Step labels cannot be duplicated between an Abnormal condition handler and its RESTART routine.

A GOTO that tries to branch across a restart keyword in either direction is a compile-time error.

An INITIATE of a restart section's parent handler from inside the restart section will cause a run-time failure; INITIATING other enabled handlers is allowed.

### RESTART heading

The crossing of a RESTART heading causes the handler to enter its RESTART routine.

### Syntax



### Description

The Restart heading is not a Preemption point.

## Accessing parameters

A TDC CL sequence program can access parameters in the MFC in which the program is loaded. It can also access some parameters in a different MFC, through the C-Link.

Parameters can be accessed in four ways: as a local variable, as a parameter of the bound Data point, as a parameter of an EXTERNAL Data point, and as a parameter of an EXTERNAL Box Data point.

### Local Variables

NUMBER and LOGICAL variables can be declared as LOCAL variables by using the AT clause to equate them with Numeric and Flag variables of the MFC.

---

|                |                                                                                                                                                                                                     |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Example</b> | <pre> LOCAL tmp AT NN(10)–default NUMBER type   LOCAL switch: LOGICAL AT FL(1)   LOCAL table: NUMBER ARRAY(10..20) AT NN(1)   ...   SET tmp = 15   IF switch = ON THEN SET table(tmp+1)= 100 </pre> |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

---

### Bound Data Point (BDP) parameters

All MFC parameters are predefined as parameters of the BDP, so that they can be used without any declarations. This method is used only to access parameters in a program's own MFC see "MC parameters accessible by TDC CL" on page 78 through "Timer parameters" on page 84.

Some of these parameters behave as Data points, having their own parameters in turn. This allows such access expressions as LP(1).SP and AI(3).ALHIFL.

---

|                |                                                                                                          |
|----------------|----------------------------------------------------------------------------------------------------------|
| <b>Example</b> | <pre> SET NN(10) = 100 SET LP(1).SP. 50 IF PF THEN FAIL IF DI(1008).PV = On THEN SET FL(05) = off </pre> |
|----------------|----------------------------------------------------------------------------------------------------------|

---

### External data point parameters

Most MFC parameters can be accessed as external Data point parameters by using the EXTERNAL declaration. This form can be used to access parameters both in the program's own MFC and in different MFCs on the same C-Link. External Data point parameter names are defined in "Analog input parameters" on page 80 through "Timer parameters" on page 84.



External Data point parameters must be in external data points defined by means of the Point Builder. Note that if a Numeric or Flag is defined as an external Data point, it is given an artificial PV parameter that contains its value.

Attempts to fetch a PV or OP parameter of a dual-digital obtains only the value of the parameter's lower subslot.

**Example**


---

```

EXTERNALA100
 - regulatory data point in this MFC
EXTERNALB100
 - regulatory point in different MFC
EXTERNALVALVE
 - digital output
EXTERNALSubSeq
 - process module in this MFC
...
SET A100.OP = 50
READ NN(10) FROM B100.PV
OPEN VALVE
 - state change INITIATE SUBSEQ: HOLD

```

---

**External box data point parameters**

Some parameters in a different MFC can be accessed as external box Data point parameters by using an EXTERNAL declaration of the predefined Box Data point name (see “Box data point identifiers” on page 22. and “External data points” on page 34).

**Example**


---

```

EXTERNAL $HY02B10
 - Box data point, Hiway 2, Box 10
...
WRITE $HY02B10.NN(5) FROM LP(10).PV
READ NN(1) FROM $HY02B10.LP(1).PV
INITIATE $HY02B10.SEQ(1)

```

---

For a TDC CL sequence program in the MFC to make a direct reference to a PV or OP of a Digital point, the code simply names the Point and parameter and uses the state text Strings as appropriate.

**MFC data point parameters**

“MC parameters accessible by TDC CL” on page 78 shows all accessible MFC parameters.

The MFC points ANALOG INPUT, ANALOG OUTPUT, DIGITAL INPUT, DIGITAL OUTPUT, LOOP, and TIMER listed in the table “MC parameters

accessible by TDC CL” on page 78 have parameters which are listed in the tables “Analog input parameters” on page 80 through “Timer parameters” on page 84.

The parameters under the SOPL Name heading marked with an \* in “Analog input parameters” on page 80 through “Timer parameters” on page 84 (for example, PV\*) are the default parameters in the SOPL environment; they could be referenced in SOPL without the parameter code (for example, READ [A1003]), where A1003 is an Analog Input point, would return the PV of A1003 by default). In TDC CL, you must specify the parameter code for all parameters you want to use (for example, READ A1003.PV). These asterisks are included in these tables as an aid to translating SOPL to TDC CL.

The parameters given in these tables also apply when the MFC parameters are defined as external Data points with user-defined names. In this case, an artificial PV parameter is added for Numeric (P in SOPL), Flag (ST in SOPL), and Logic block (ST in SOPL) Data points.

Digital and Analog composite points are listed in “MC parameters accessible by TDC CL” on page 78 as if the input and output components were separate Points. To find Composite point parameters, look in the separate parameter lists for the equivalent Input or Output point.



#### Caution

MC Data Point Flag Variables (PV) 1 through 64 have additional alarming functionality that does not exist for the other flag PVs (65 through 256). Changing any of the first 64 flag variables at a fast frequency (for example, with CL/MC statements such as **SET FL(64) = ON**, and later, **SET FL(64) = OFF**) could cause exception scans that overload the channel communications. Flags 65 through 256 should be used as boolean variables when alarming is not required.

**Table 24** MC parameters accessible by TDC CL

| Parameter name | SOPL name | Data type        | Array bounds *        | Description    | Remarks          |
|----------------|-----------|------------------|-----------------------|----------------|------------------|
| ADFL           | PFn       | Logical          |                       | A/D slot fail  | View only        |
| AI             | AI        | Analog input**   | 1-16(p)               | Analog input   | p=1-16***        |
| AO             | AO        | Analog output**  | 11-164<br>(s*10+p)    | Analog output  | s=1-16 p=1-4***  |
| CI             | CI        | Counter input**  | 11-164<br>(s*10+p)    | Counter input  |                  |
| DI             | DI        | Digital input**  | 101-1616<br>(s*100+p) | Digital input  | s=1-16 p=1-16*** |
| DO             | DO        | Digital output** | 11-168<br>(s*10+p)    | Digital output | s=1-16, p=1-8*** |

**Table 24** MC parameters accessible by TDC CL (*continued*)

| Parameter name | SOPL name | Data type | Array bounds* | Description     | Remarks               |
|----------------|-----------|-----------|---------------|-----------------|-----------------------|
| FL             | FL        | Logical   | 1-256         | Flag            |                       |
| IOADFL         | PFn       |           |               | I/O A/D fail    | View only             |
| IOF1FL         | PFn       |           |               | I/O file 1 fail |                       |
| IOF2FL         | PFn       |           |               | I/O file 2 fail |                       |
| IOFL           | PFn       |           | 1-16          | I/O slot fail   |                       |
| LB             | LB        |           | 1-128         | Logic block     |                       |
| LP             | LP        | Loop**    | 1-16 (s)      | Regulatory ctrl | s=1-16***             |
| LPFL           | PFn       | Logical   | 1-16          | Reg. slot fail  | View only             |
| NN             | NN        | Number    | 1-160         | Numeric         |                       |
| OUT1FL         | PFn       | Logical   |               | OUT card 1 fail | View only             |
| OUT2FL         | PFn       |           |               | Out card 2 fail |                       |
| PRTFL          | PF        |           |               | Partial fail    |                       |
| SEQ            |           |           | 1-16          | Process module  | Used only in INITIATE |
| SFTFL          | SF        |           |               | Soft fail       | View only             |
| TM             | TM        | Timer**   | 1-32          | Timer           |                       |
| TOG            | TG        | Logical   |               | Time-out gate   | View only             |

\* Array index must be an integer constant. If array bounds are not given, the parameter is scalar.

\*\* These MFC parameters have parameters; see “Analog input parameters” on page 80 through “LOOP (Regulatory or Monitoring control slot) parameters” on page 82. The names here match the names of “Analog input parameters” on page 80 through “LOOP (Regulatory or Monitoring control slot) parameters” on page 82.

\*\*\* **s** means Slot number and **p** means Point number. Only valid Point and Slot numbers are allowed.

**Table 25** Analog input parameters

| Parameter name * | SOPL name | Data type | Description           | Remarks   |
|------------------|-----------|-----------|-----------------------|-----------|
| ALHIFL           | AH        | Logical   | Hi alarm flag         | View only |
| ALLOFL           | AL        |           | Lo alarm flag         |           |
| AV               | AV        | Number    | Accum. value (EU)     |           |
| DEV              | DV        |           | Deviation (%)         |           |
| DEVHITP          | DH        |           | Dev Hi Trip point (%) |           |
| DEVLOTP          | DL        |           | Dev Lo trip point (%) |           |
| PRTFL            | PF        | Logical   | Partial failure       | View only |
| PTINAL           | AS        |           | Point in alarm        |           |
| PV               | PVE       | Number    | PV (EU)               |           |
| PVHITP           | PHE       |           | PV Hi Trip point (EU) |           |
| PVLOPT           | PLE       |           | PV Lo Trip point (EU) |           |
| PVP              | PV*       |           | PV (%)                | View only |
| PVPHITP          | PH        |           | PV Hi Trip point (%)  |           |
| PVPLOTP          | PL        |           | PV Lo Trip point (%)  |           |
| RESETCMD         |           | Enum **   |                       |           |
| SP               | SPE       | Number    | SP (EU)               |           |
| SPP              | SP        |           | SP (%)                |           |
| STATE            | ST        | Enum **   | Accum.state           | View only |
| STRTSTOP         | RS        | Enum ***  | Accum.restart         |           |

\* See Note 1. on page 84.

\*\* See Note 2. on page 85.

\*\*\* See Note 3. on page 85.

**Table 26** Analog output parameters

| Parameter name * | SOPL name | Data type  | Description           | Remarks   |
|------------------|-----------|------------|-----------------------|-----------|
| MODATTR          |           | Enum **    | Mode attribute        | View only |
| MODE             | MD        | MCMODE *** | MODE                  |           |
| NMODATTR         |           | Enum **    | Normal mode attribute | View only |
| NMODE            |           | MCMODE *** | Normal mode           |           |

**Table 26** Analog output parameters *(continued)*

| Parameter name * | SOPL name | Data type | Description      | Remarks   |
|------------------|-----------|-----------|------------------|-----------|
| OP               | OV        | Number    | Output value (%) |           |
| PRTFL            | PF        | Logical   | Partial failure  | View only |

\* See Note 1. on page 84.

\*\* See Note 2. on page 85.

\*\*\* See Note 3. on page 85.

**Table 27** Counter input parameters

| Parameter name * | SOPL name | Data type | Descriptions         | Remarks   |
|------------------|-----------|-----------|----------------------|-----------|
| AV               | AV        | Number    | Accum.value          | View only |
| PDEVFL           | AL        | Logical   | Pre-alarm flag       |           |
| PDEVTP           | SL        | Number    | Pre-alarm Trip point |           |
| PRESET           | SH        |           | Alarm limit          |           |
| PRESETFL         | AH        | Logical   | Alarm flag           | View only |
| PRTFL            | PF        |           | Partial failure      |           |
| PV               | PV        | Number    | Current pulse        |           |
| RESETCMD         | -         | Enum**    |                      |           |
| STATE            | ST        |           | Counter state        | View only |
| STRTSTOP         | RS        |           | Counter restart      |           |

\* See Note 1. on page 84.

\*\* See Note 2. on page 85.

**Table 28** Digital input parameters

| Parameter name * | SOPL name | Data type | Description    | Remarks   |
|------------------|-----------|-----------|----------------|-----------|
| PRTFL            | PF        | Logical   | Partial fail   | View only |
| PTINAL           | AS        |           | Point in alarm |           |
| PV               | ST*       |           | Input state    |           |

\* See Note 1. on page 84.

**Table 29** Digital output parameters

| Parameter name <sup>*</sup> | SOPL name       | Data type             | Description     | Remarks   |
|-----------------------------|-----------------|-----------------------|-----------------|-----------|
| MODATTR                     |                 | Enum <sup>**</sup>    | Mode attribute  | View only |
| MODE                        | MD              | MCMODE <sup>***</sup> | Mode            |           |
| NMODATTR                    |                 | Enm <sup>**</sup>     | Normal mode     | View only |
| NMODE                       |                 | MCMODE <sup>***</sup> | Normal mode     |           |
| OP                          | ST <sup>*</sup> | Logical               | Output state    |           |
| PRTFL                       | PF              |                       | Partial failure | View only |
| PV                          | FB              |                       | Feedback input  |           |

<sup>\*</sup> See Note 1. on page 84.

<sup>\*\*</sup> See Note 2. on page 85.

<sup>\*\*\*</sup> See Note 3. on page 85.

**Table 30** Logic block parameters

| Parameter name <sup>*</sup> | SOPL name | Data type | Description   | Remarks   |
|-----------------------------|-----------|-----------|---------------|-----------|
| PV                          |           | Logical   | Current state | View only |

<sup>\*</sup> See Note 1. on page 84.

**Table 31** LOOP (Regulatory or Monitoring control slot) parameters

| Parameter name <sup>*</sup> | SOPL name | Data type | Description           | Remarks   |
|-----------------------------|-----------|-----------|-----------------------|-----------|
| ALHIFL                      | AH        | Logical   | Hi alarm flag         | View only |
| ALLOFL                      | AL        |           | Lo alarm flag         |           |
| BIAS                        | BS        | Number    | Bias                  |           |
| DEV                         | DV        |           | Deviation (%)         | View only |
| DEVHITP                     | DH        |           | Dev Hi Trip point (%) |           |
| DEVLOTP                     | DL        |           | Dev Lo Trip point (%) |           |
| INITMAN                     | IM        | Logical   | IM mode               | View only |
| K                           | KP        | Number    | K on PID              |           |
| K1                          | K1        |           | Aux gain on X         |           |
| K2                          | K2        |           | Aux CP bias (%)       |           |
| KA                          | KA        |           | Aux gain on Y         |           |

**Table 31** LOOP (Regulatory or Monitoring control slot) parameters *(continued)*

| Parameter name * | SOPL name | Data type | Description           | Remarks   |
|------------------|-----------|-----------|-----------------------|-----------|
| LM               | LM        | Logical   | LM mode               | View only |
| MODATTR          |           | Enum**    | Mode attribute        |           |
| MODE             | MD        | MCMODE*** | Mode                  |           |
| NMODATTR         |           | Enum**    | Normal mode attr      | View only |
| NMODE            |           | MCMODE*** | Normal mode           |           |
| OAUTO            | AUT       | Logical   | AUTO mode             |           |
| OCAS             | CAS       |           | CASC mode             |           |
| OCOM             | COM       |           | COMP mode             |           |
| OMAN             | MAN       |           | MAN mode              |           |
| OP               | OV        | Number    | Output value (%)      |           |
| OPHILM           | OH        |           | OP Hi limit (%)       |           |
| OPLOLM           | OL        |           | OP Lo limit (%)       |           |
| OPU              | OVX       |           | OP unlimited (%)      |           |
| PAUTO            | SAU       | Logical   | SEQ-AUTO mode         |           |
| PCAS             | SCA       |           | SEQ-CASC mode         |           |
| PCOM             | SCO       |           | SEQ-COMP mode         |           |
| PMAN             | SMA       |           | SEQ-MAN mode          |           |
| PROG             | SEQ       |           | SEQ mode              | View only |
| PROGVAL          | SQV       | Number    | SEQ input (%)         |           |
| PRTFL            | PF        | Logical   | Partial Failure       | View only |
| PTINAL           | AS        |           | Point in alarm        |           |
| PV               | PVE       | Number    | PV (EU)               |           |
| PVEUHI           | RG1       |           | RANGE 100% (EU)       |           |
| PVEULO           | RG0       |           | Range 0% (EU)         |           |
| PVHITP           | PHE       |           | PV Hi Trip point (EU) |           |
| PVLOTP           | PLE       |           | PV Lo Trip point (EU) |           |
| PVP              | PV*       |           | PV (%)                | View only |
| PVPHITP          | PH        |           | PV Hi Trip point (%)  |           |
| PVPLOTP          | PL        |           | PV Lo Trip point (%)  |           |
| PVRAW            | APV       |           | Analog PV             | View only |
| PVROCNTP         | CL        |           | ROC - Trip point (%)  |           |
| PVROCPTP         | CH        |           | ROC + Trip point (%)  |           |
| RATIO            | RT        |           | Ratio                 |           |

**Table 31** LOOP (Regulatory or Monitoring control slot) parameters *(continued)*

| Parameter name <sup>*</sup> | SOPL name | Data type | Description      | Remarks |
|-----------------------------|-----------|-----------|------------------|---------|
| SOA                         | OA        | Logical   | Status out (SOA) |         |
| SOB                         | OB        |           | Status out (SOB) |         |
| SP                          | SPE       | Number    | SP (EU)          |         |
| SPHILM                      | SHE       |           | SP Hi limit (EU) |         |
| SPLOLM                      | SLE       |           | SP Lo limit (EU) |         |
| SPP                         | SP        |           | SP (%)           |         |
| SPPHILM                     | SH        |           | SP Hi limit (%)  |         |
| SPPLOLM                     | SL        |           | SP Lo limit (%)  |         |
| T1                          | T1        |           | TI on PID        |         |
| T1L                         | T1L       |           | Lead time        |         |
| T2                          | T2        |           | TD on PID        |         |
| T2L                         | T2L       |           | Lag time         |         |

\* See Note 1. on page 84.

\*\* See Note 2. on page 85.

\*\*\* See Note 3. on page 85.

**Table 32** Timer parameters

| Parameter name <sup>*</sup> | SOPL name       | Data type              | Description         | Remarks   |
|-----------------------------|-----------------|------------------------|---------------------|-----------|
| PV                          | PV              | Number                 | Current time        | View only |
| RESETCMD                    |                 | Enum                   |                     |           |
| SP                          | SP              | Number                 | Setting time (secs) |           |
| SPMIN                       |                 |                        | Setting time (mins) |           |
| STATE                       | ST <sup>*</sup> | Enum <sup>**</sup>     | Timer state         | View only |
| STRTSTOP                    | RS              |                        | Timer restart       |           |
| TIMELEFT                    | RV              | Number                 | Remaining time      | View only |
| TIMOUTFL                    | AS              | Logical <sup>***</sup> | Time-out flag       |           |

\* See Note 1. on page 84.

\*\* See Note 2. on page 85.

\*\*\* See Note 4. on page 86.

## Notes

1. The parameters for this table are scalar.



2. Enumeration parameters have the following data types:

| Parameter Name    | Enumeration Name | Enumeration States |
|-------------------|------------------|--------------------|
| MODATTR, NMODATTR | MODATTR          | OPERATOR/PROGRAM   |
| STATE             | STATE            | STOPPED/RUNNING    |
| RESETCMD          | RESET            | RESET/RESTSTRT     |
| STRTSTOP          | STRTSTOP         | STOP/START         |

Reading of RESET or STRTSTOP parameters returns RESET or STOP value if STATE = STOPPED, and RESTSTRT or START if STATE = RUNNING.

A variable or subroutine argument that was declared with a state name list as its data type is not compatible with variables, arguments or parameters whose data type is in the above list.

This is because the TDC CL compiler uses the LCN view of system-defined Enumeration types, all of which have more states than the MFC view of these types described above. For example a local variable declared as:

```
LOCAL var1: stopped/running at FL(01)
```

cannot be used in a set statement with the state parameter of a timer as in:

```
SET Var1 =TM(01).STATE
```

If var1 was declared as:

```
Local var1 :STATE at FL(01)
```

The SET statement would be accepted by the compiler.

3. Parameters MODE and NMODE are the enumeration type MCMODE, which has the following states:

| Parameter Name | Enumeration Name | Enumeration States                                                 |
|----------------|------------------|--------------------------------------------------------------------|
| MODE, NMODE    | MCMODE           | OMAN/OAUTO/OCAS/OCOM/<br>PMAN/PAUTO/PCAS/PCOM/<br>MAN/AUTO/CAS/COM |

Do not confuse this with the type MODE, which is available on other LCN-connected boxes and which has two different states.

The values MAN, AUTO, CAS, and COM are used only for storing into MODE or NMODE; they can not be compared with MODE or NMODE.

The values OAUTO, OCAS, PAUTO, PCAS, and CAS cannot be stored into or compared with MODE or NMODE of Digital Output, Digital Composite, Analog Output, or Analog Composite data points.

Reading of MODE (NMODE) parameter returns OMAN, OAUTO, OCAS, or OCOM values if MODATTR/NMODATTR = OPERATOR and PMAN,

PAUTO, PCAS, or PCOM values if MODATTR/NMODATTR = PROGRAM.

4. The LCN view of the parameter TIMOUTFL is Time Up/Normal. The MFC view is Off/On.

The following are examples of gaining access to MFC Parameters:

|                     |                               |
|---------------------|-------------------------------|
| EXTERNAL My_AO      | Analog output point           |
| SET My_AO.OP = 5    | Parameter of external point   |
| SET AO(11).OP = 5   | Parameter of BDP parameter    |
| EXTERNAL My_NUM     | Numeric data point            |
| SET My_NUM.PV = 100 | Parameter of external point   |
| SET NN(15) = 100    | Parameter of bound data point |
| SET My_NUM = 100    | INVALID                       |

### Self-defining enumeration

Some parameters of MFC points are defined as type Logical if accessed as parameters of the BDP, but if accessed as parameters of an EXTERNAL, the parameters are defined as self-defining Enumerations. This applies to the following parameters:

- PV of Digital Input data point
- PV and OP of Flag data point
- PV of Digital Composite point
- OP of Digital Composite point
- PV of Logic Block point

The state names for the self-defining Enumeration are found in the STATE1 and STATE2 parameters of the EXTERNAL data point.

---

#### Example

```
EXTERNAL VALVE-DO point, OPEN/CLOSE
EXTERNALr (relop)-DI point, OK/NG
EXTERNAL FULLFL-Flag point, FULL/EMPTY
...
SET VALVE.OP = CLOSE
OPEN VALVE
IF SWITCH.PV = OK THEN SET FULLFL.PV = EMPTY
SET FULLFL.PV = FULL
```

---

## User-written subroutines

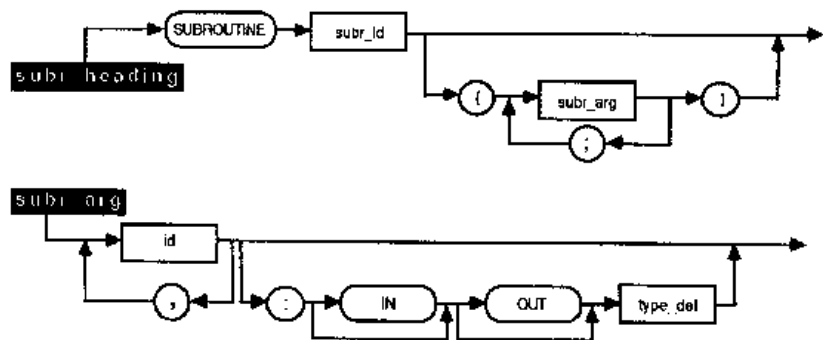
This section describes user-written TDC CL subroutines. A program can call subroutines that are user-written or system-supplied: user-written subroutines must use the facility described here, but system-supplied subroutines need not be written in TDC CL.

User-written TDC CL subroutines are referred to as Sequence subroutines. See “General” on page 68.

### SUBROUTINE heading

The SUBROUTINE heading identifies it and specifies its arguments, including their type and access mode.

### Syntax



### Description

Each argument has an access mode: IN, OUT, or IN OUT. An argument's mode determines whether the subroutine can access that argument, set it, or both. IN arguments can only be accessed. OUT arguments can only be set. IN OUT arguments can be both accessed and set.

If an arguments mode is omitted, the default is IN. The default for all optional type identifiers is Number.

### Example

```
SUBROUTINE test (x, y: Number;
b: OUT Logical c;IN OUT)
 IF x = y THEN SET b = on
 - valid
 IF c 44.4 THEN SET c = 44.4
```

```
 - valid
SET x = y
 - NOT valid, x is IN by default
 IF b THEN SET c = y
 - NOT valid, b is OUT
END test
```

---

## Arguments

The following data types can be used as arguments in a subroutine:

- Number
- Logical
- Enumeration
- State name list
- Arrays of Number
- Arrays of Logical

Data point identifiers are not permitted as subroutine arguments.

Enumeration types that can be passed as subroutine arguments are limited as follows: state name lists can contain only two states and enumerations are limited to those shown in “Analog input parameters” on page 80 through “LOOP (Regulatory or Monitoring control slot) parameters” on page 82.

## Built-in functions and subroutines

### Arithmetic functions

If an enumeration is called for by a parameter list file or an External declaration and that enumeration has a state name that is the same as a built-in subroutine or function, the compiler does not let the enumeration be declared. A parameter of that enumeration type cannot be referenced in a TDC CL program.

All functions listed below accept arguments of type Number, and return Number results; in addition, the trigonometric functions listed below *must be specified in radians, not degrees*.

The built-in function ROUND does not work correctly on negative numbers in the MFC; write in-line code or a subroutine that will perform the ROUND function.

|                 |                                |
|-----------------|--------------------------------|
| Abs (x)         | absolute value                 |
| Exp (x)         | exponential                    |
| Int (x)         | truncate to integer            |
| Ln (x)          | natural logarithm              |
| Log10 (x)       | common logarithm               |
| Max (x, y, ...) | maximum (5 or fewer arguments) |
| Min (x, y, ...) | minimum (5 or fewer arguments) |
| Round (x)       | round to integer               |
| Sqrt (x)        | square root                    |

### Set\_Time subroutine

The built-in subroutine Set\_Time is provided to start a timer. The timer must be in the same MFC as the program calling Set\_Time. Its syntax is:

```
Set_Time (x: a timer point;
& count: Time;
& start: Logical)
```

This subroutine sets the timer x to count for count time. If start is equal to On, the timer is started; otherwise, it is left inactive.

The Count specification cannot be a generalized Time expression, but is restricted as in the WAIT statement; therefore it can be one of:

```
arith_operand MINS
arith_operand SECS
```

If the (truncated) value of the arith\_operand is less than zero or greater than 9999, 0 or 9999, respectively, is used instead.

---

|                |                                     |
|----------------|-------------------------------------|
| <b>Example</b> | CALL SET_TIME (TM(01), 15 SECS, ON) |
|----------------|-------------------------------------|

---

**Day\_Time function**

The syntax of the Function Day\_Time is

Day\_Time: Logical(H, M : Number)

Day\_Time returns On if the current time of day is H hours M minutes.

---

|                |                                           |
|----------------|-------------------------------------------|
| <b>Example</b> | IF DAY_TIME (16,30) THEN GOTO PHASE BEGIN |
|----------------|-------------------------------------------|

---

---

## Syntax summary

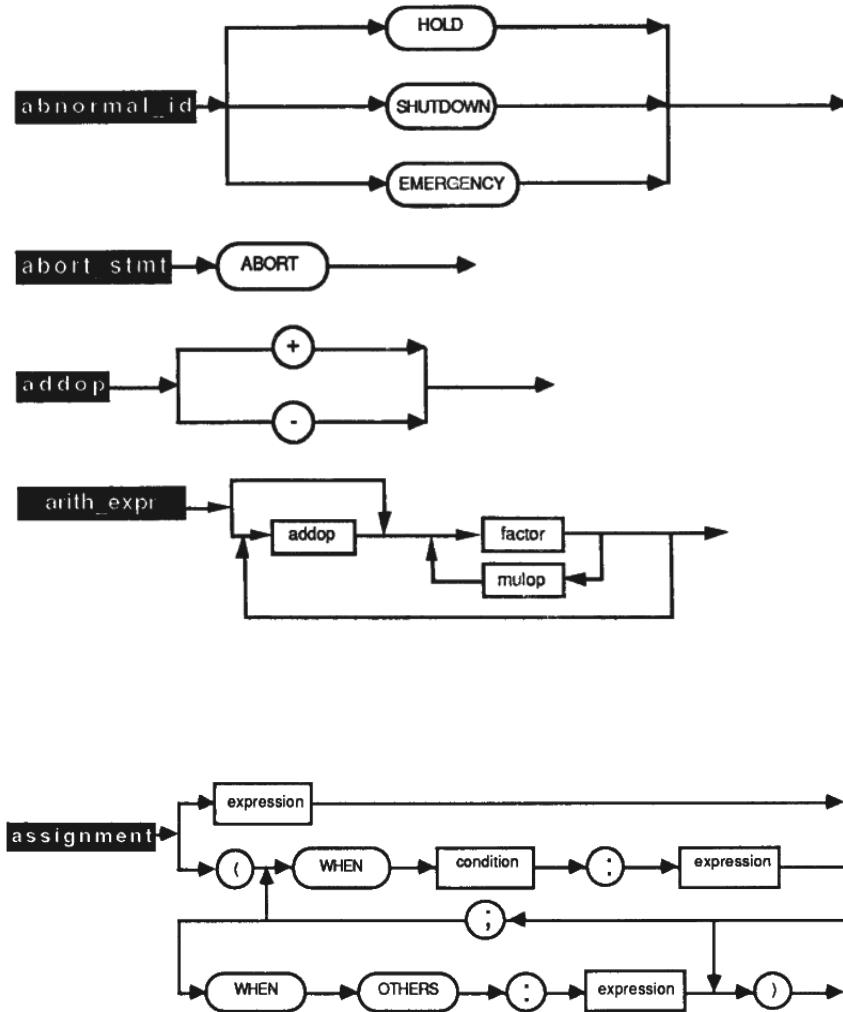
This topic provides a quick reference to TDC CL syntax. It is a summary of the rules of form for TDC CL, and is divided into two parts:

- The first part is a summary of TDC CL syntax in the form of all the syntax diagrams that were presented throughout the manual. Each syntax diagram is labeled (in reverse-video), and they are arranged in alphabetical order.
- The second part is a summary of TDC CL syntax in the BNF form of production rules, again in alphabetical order.

To use either part of this topic, decide what item you want to construct and go through either summary (depending on what form you feel most comfortable with) looking for that item on the left-most portion of each page. When you find the item, the way to build it is contained in the diagram or production rule to the right of the item. Note that some simple items that are listed individually in the BNF version (“Notation used for syntax production rules” on page 102) are included within more complex diagrams.

## Syntax diagram summary

Figure 8 Abnormal\_id - Arith\_expr





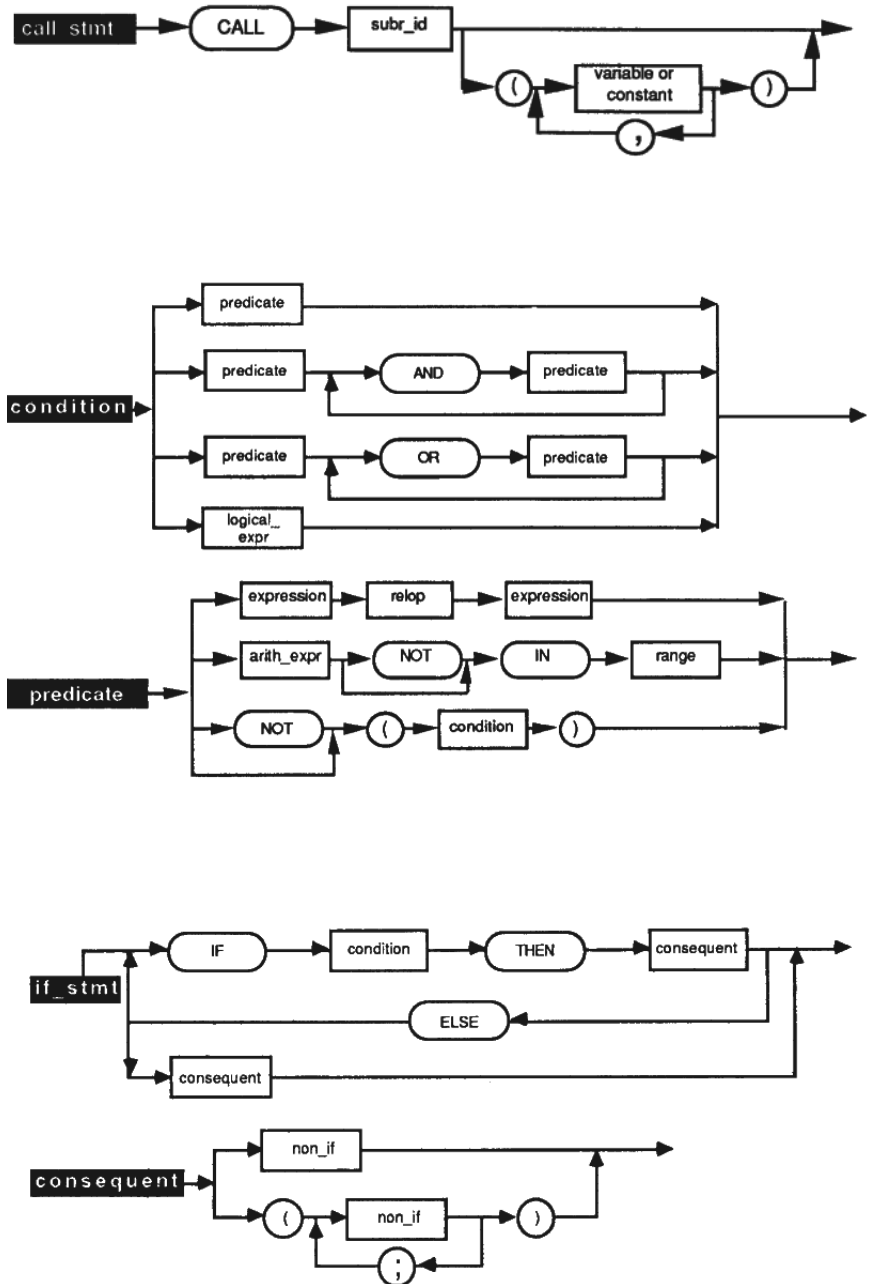


Figure 9 \assignment - Declaration

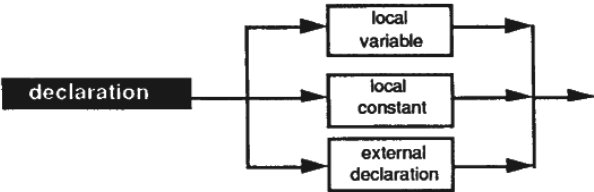


Figure 10 End\_statement - Goto\_stmt

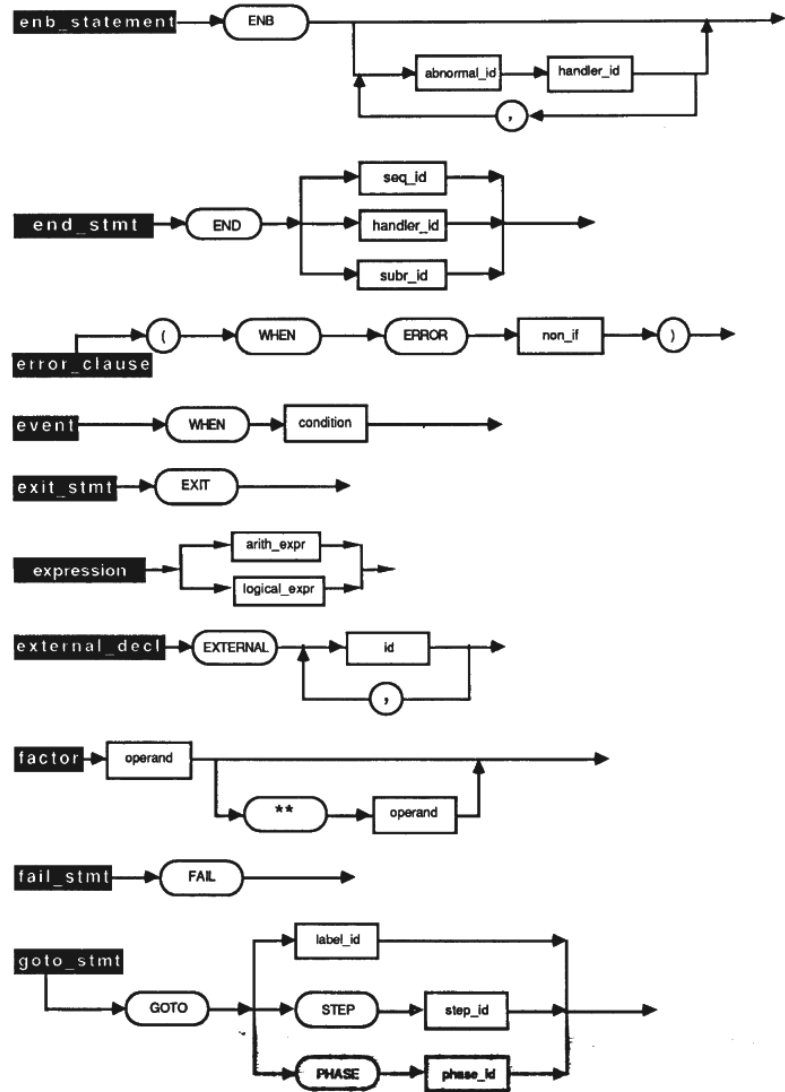


Figure 11 Handler - Initiate\_stmt

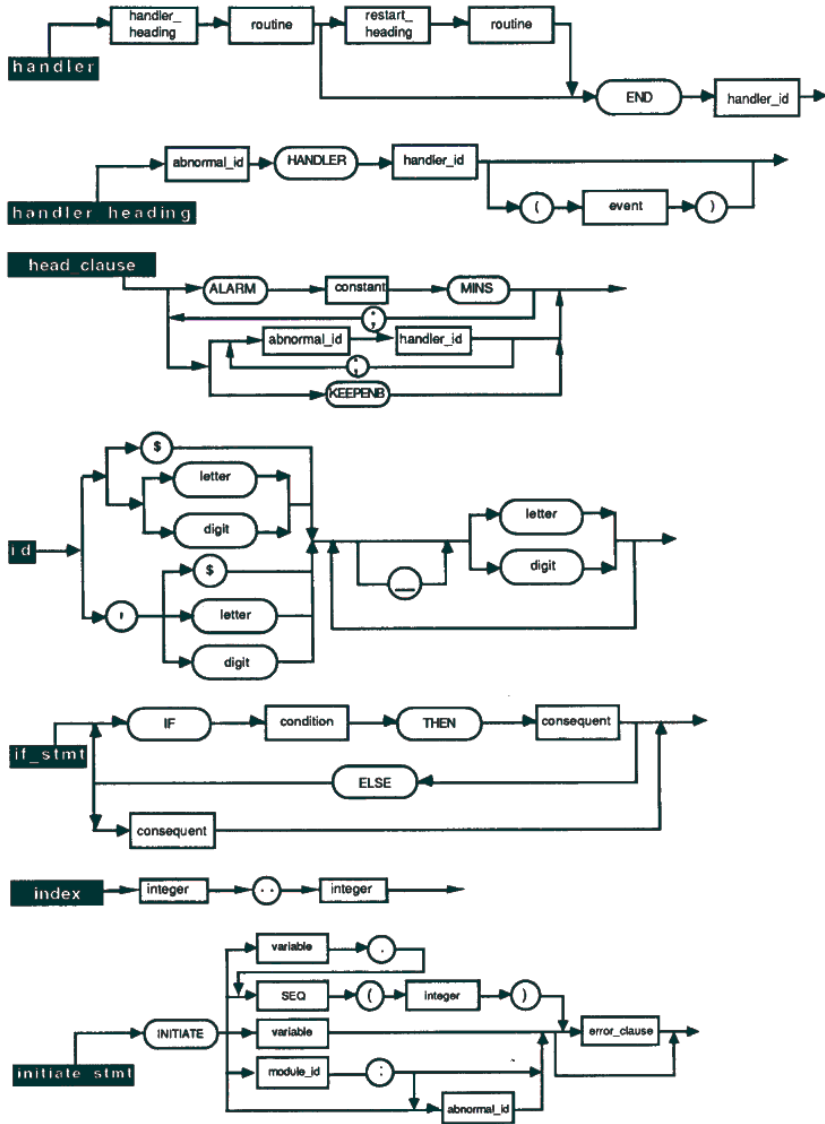


Figure 12 Integer - Main\_sequence

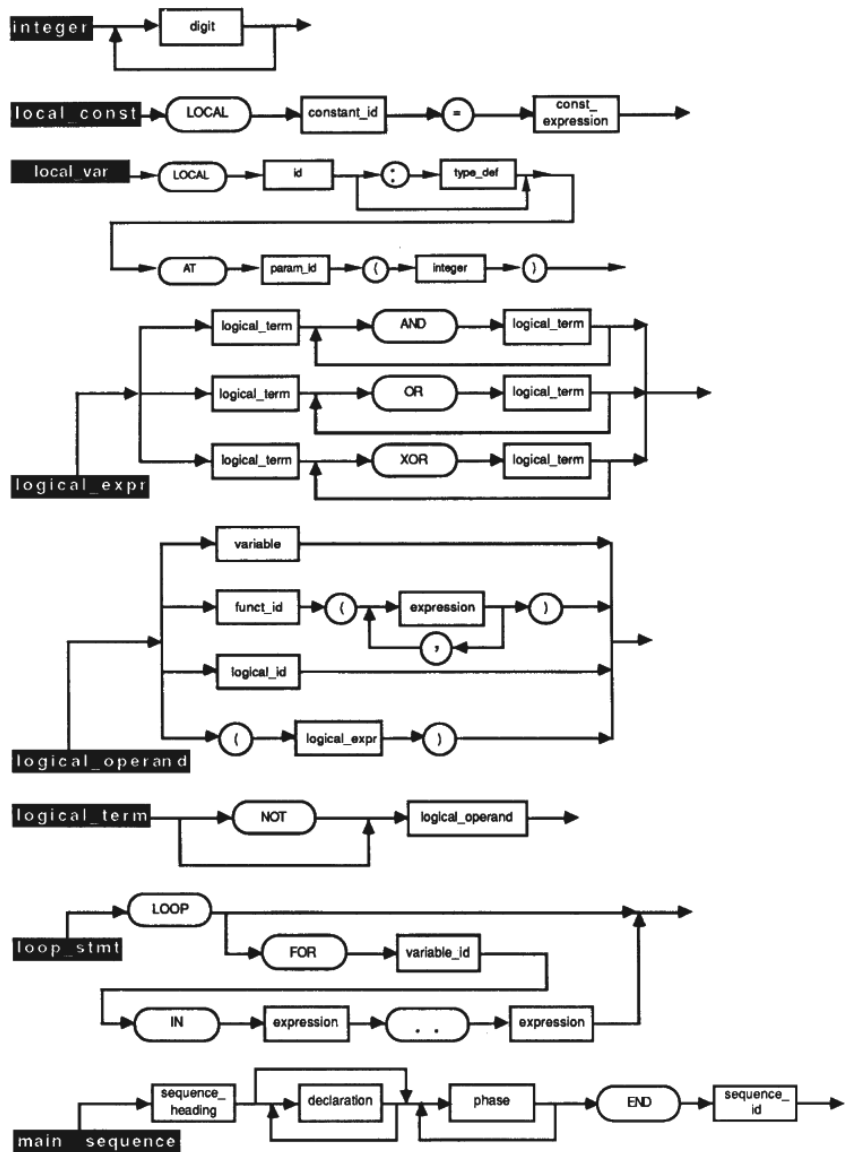




Figure 14 Send\_stmt - Step\_heading

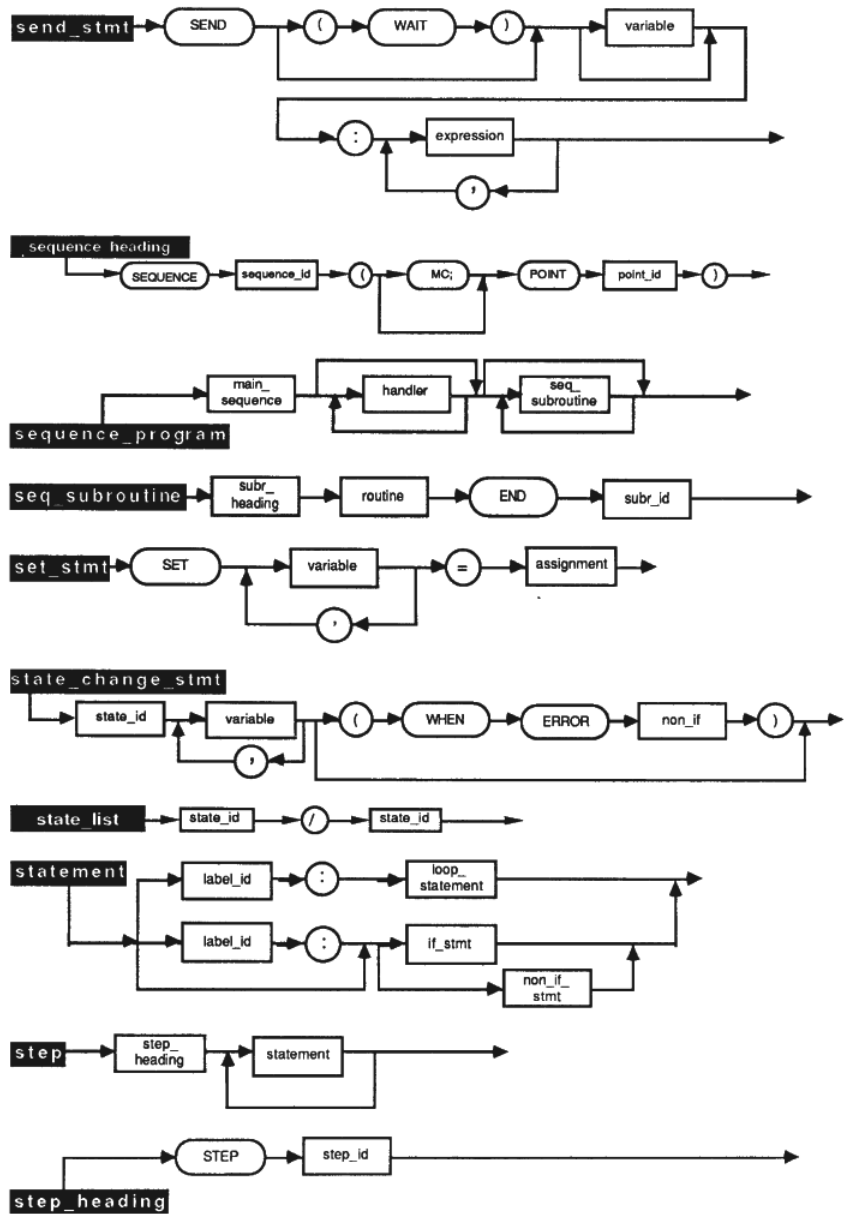


Figure 15 Suger\_arg - Wait\_stmt

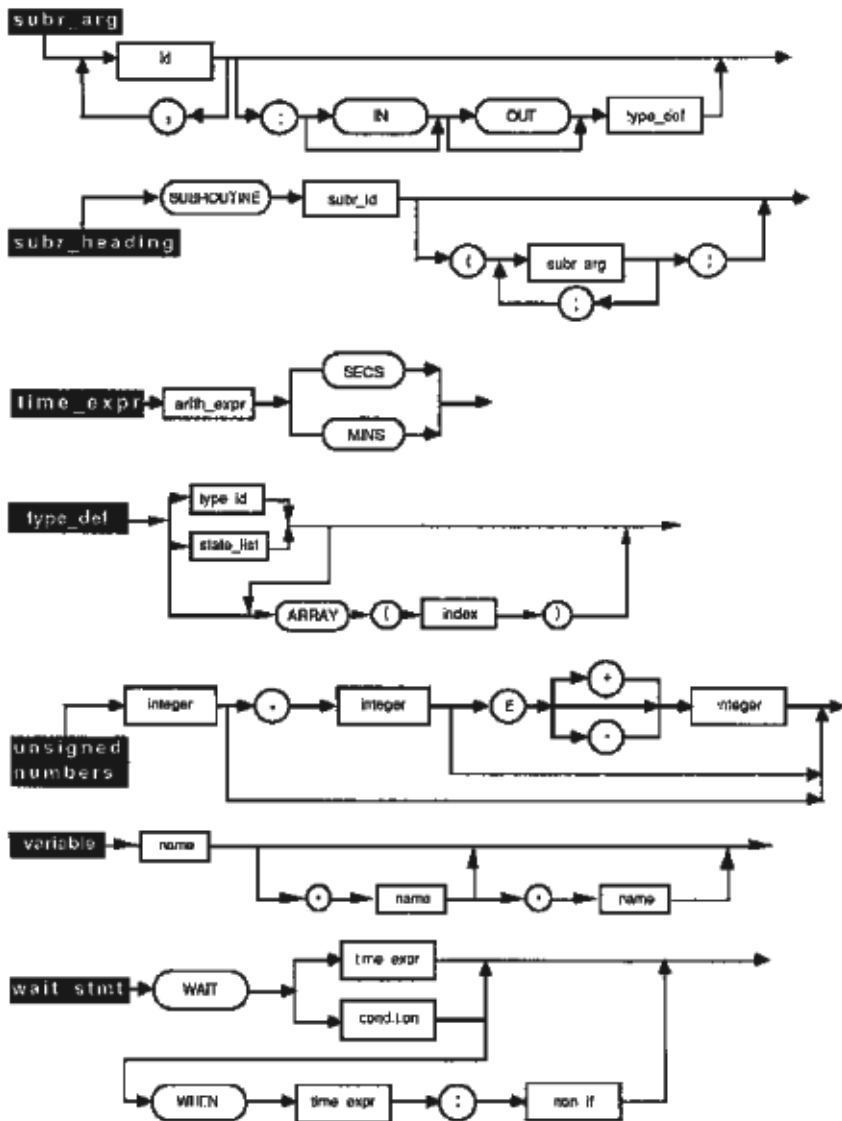
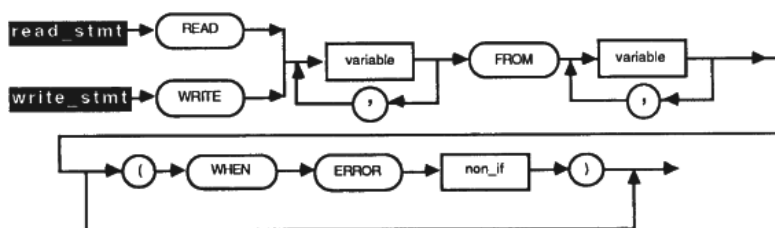




Figure 16 Read\_stmt - Write\_stmt



## Notation used for syntax production rules

The following presentation of TDC CL syntax effectively follows BNF notation conventions as follows:

Sequences of lower-case characters and embedded underscores mean things are to be combined according to the exact form expressed under this Syntax heading.

Upper-case characters and special characters appear as written, except for the symbols `::=`, `{`, `}`, `[`, `]`, and `|`, which are described as follows.

- An item enclosed in braces (`(, )`) stands for the occurrence of that item zero or more times.
- An item enclosed in square brackets (`[, ]`) stands for the occurrence of that item zero or one times (that is, the item is optional).

The symbols `::=` and `|` stand for production (how to build, or put together) and alternation, respectively. For example, `x ::= y | z` can be read `x` produces `y` or `z`. Another way to explain the `::=` symbol is that in order to form `x`, `y` or `z` must be present in the form listed. Many times, a form on the right of the `::=` symbol is itself given a syntactic form, as follows: using the same example, `x ::= y | z`. A further rule that governs the form of `z` is specified, indented and just below the form that specifies how to form `x`. For example:

```
x ::= y | z
y ::= point_param_sp
z ::= point_param_pv
```

This means that to produce `x` you need to specify `y` or `z`; `y` is produced by specifying a Point's Setpoint, and `z` is produced by specifying a Point's Process variable parameter PV.

Unless otherwise noted, all symbols ending in `_ID` are ordinary identifiers (that is, `anything_ID ::= ID`).

## Production rules

```

abnormal_ID ::= HOLD | SHUTDOWN | EMERGENCY
abort_stmt ::= ABORT
addop ::= + | -
arith_expr ::= [addop] term {addop term}
array_def ::= ARRAY (index)
assignment ::= expression
 | (WHEN condition : expression
 { ; WHEN condition : expression }
 [; WHEN OTHERS : expression])
call_stmt ::= CALLsubr_ID [(expression)]
compilation_unit ::= sequence_program
 | lib_subroutine
condition ::= predicate {AND predicate}
 | predicate {OR predicate}
 | logical_expr
consequent ::= non_if
 | (non_if {; non if})
declaration ::= local_var
 | local_const
 | external_decl
digit ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
else_if_stmt ::= ELSE IF condition THEN consequent
else_stmt ::= ELSE consequent
enb_statement ::= ENB [abnormal_id_handler id {,abnormal_id
 handler_id}]
end_statement ::= END_seq_id | END handler_id | END subr_id
error_clause ::= (WHEN ERROR non_if)
event ::= WHEN condition
exit_stmt ::= EXIT
expression ::= arith_expr | logical_expr
expressions ::= expression {, expression}
external_decl ::= EXTERNAL id {, id}
factor ::= operand [** operand]
fail_stmt ::= FAIL
first_char ::= $ | letter
firs_param ::= point_description | subr_arg
funct_args ::= (typed_IDs (; typed_IDs))
goto_stmt ::= GOTO label_ID
 | STEP step_ID | PHASE phase_ID
handler ::= handler_heading
 routine
 [restart_heading
 routine]
 END handler_ID
head_clause ::= [ALARM constant MINS]
 | [ALARM constant MINS ;] KEEPENB
 | [ALARM constant MINS ;] abnormal_ID handler_ID
 {;abnormal_ID handler_ID}
help_attribute ::= HELP String

```

```

ID ::= first_char ([_] letter or digit)
 | ' special first char { [_] letter_or_digit }
IDs ::= ID{,ID}
if_stmt ::= IF condition THEN consequent
 { else_if_stmt }
 [else_stmt]
index ::= integer..integer
initiate_stmt ::= INITIATE SEQ (integer) [error_clause]
 | INITIATE [variable] [error_clause]
 | INITIATE [module_ID:] abnormal_ID [error_clause]
 | INITIATE variable.SEQ (integer) [error_clause]
integer ::= digit{digit}
letter ::= upper_case_alphabetic
 | lower_case_alphabetic
 | Katakana
letter_or_digit ::= letter | digit
local_const ::= LOCAL constant_ID = constant_decl
local_var ::= LOCAL id [: type_def AT param_ID (integer)
logical_expr ::= logical_term {AND logical_term}
 | logical_term {OR logical_term}
 | logical_term {XOR logical_term}
logical_operand ::= variable
 | function_ID (expressions)
 | logical_ID
 | (logical_expr)
logical_term ::= [NOT] logical_operand
loop_def ::= LOOP [FOR variable_ID IN range]
loop_stmt ::= LOOP [FOR variable_ID IN range]
main_sequence ::= sequence_heading
 { declaration }
 phase
 { phase }
 END sequence_ID
mulop ::= * | / | MOD
name ::= ID | array_ID (arith_expr)
non_if ::= set_stmt | read_stmt | write_stmt
 | state_change_stmt|goto_stmt | repeat_stmt
 | pause_stmt| wait_stmt| call_stmt
 | send_stmt| receive_stmt| initiate_stmt
 | fail_stmt| resume_stmt| exit_stmt
 | abort_stmt| enb_stmt| end_stmt
one_d_array ::= (const_expression{,const_expression})
operand ::= variable
 | function_ID (expressions)
 | (arith_expr)
 | unsigned_number
 | constant_ID
pause_stmt ::= PAUSE
phase ::= phase_heading
 routine

```

```

phase_heading ::= PHASE phase_ID[(head_clause
 {;head clause})]
predicate ::= expression relop expression
 | arith_expr [NOT] IN range
 | [NOT] (condition)
program ::= sequence_program
quoted String ::= "String"
read_stmt ::= READ variables FROM variables [error_clause]
relop ::= <| = |>| <= | <> | > =
repeat_stmt ::= REPEAT target
restart_heading ::= RESTART
resume_stmt ::= RESUME PHASE phase_ID
routine ::= step {step} | statements
schedule ::= event
 | time_expr
 | WAIT
send_stmt ::= SEND [(WAIT)] [variable] : expressions
sequence_heading ::= SEQUENCE sequence_ID ([MFC ;] POINT point_
id)
sequence_program ::= main_sequence { handler }
 { seq_subroutine_}
 (see subroutine syntax)
seq_subroutine ::= subr_heading
 routine
 END subr_ID
set_stmt ::= SET variables = assignment
sign::=+ | -
simple_type ::= type_ID | state_list
special_first_char ::= $ | letter | digit
state_change_stmt ::= state_ID [variables]
 [error_clause]
state_list ::= state_ID/state_ID
statement ::= label ID : loop_stmt | [label_ID :] unlabeled_stmt
statements : : = statement { statement }
step ::= step_heading {statements}
step_heading ::= STEP step_ID
String ::= " (String_chr) "
string_chr ::= any ASCII character_except_quote | ""
subr_arg ::= IDs [: [IN] [OUT] Simple_type]
subr_args ::= (first_param {; subr_arg})
subr_heading ::= SUBROUTINE subr_ID [subr_args]
term ::= factor {mulop factor}
time_expr ::= arith_expr SECS | arith_expr MINS
type_def ::= simple_type | [type_ID] array_def
type_IDs ::= IDs[:type_def]
unlabeled_stmt ::= if_stmt | non_if
unsigned_number ::= integer[.integer[E[sign]integer]]
variable ::= name
 | name . name
 | name . name . name
variables ::= variable {, variable}

```

```
wait_stmt ::= WAIT time_expr
 | WAIT condition [(WHEN time_expr:non_if]
write_stmt ::= WRITE variables FROM variables
 [error_clause]
```

# Index

## C

- commands
  - basic and extended controller 4
  - Data Hiway switch 4
  - MC 5
  - PIU, DHP and HIM 5
  - reserve controller 5
  - TDC Data Hiway 4

## D

- Data Hiway
  - switch command 4

## E

- error code
  - UAC Reserve Controller MC error codes
    - Reported by the RCD 49
- error codes
  - Basic Controller Primary (1xx) and Reserve (6xx) 21
  - DHP or 620 HIM 17
  - Extended Controller Primary (7xx) and Reserve (8xx) 23
  - Hiway addressing/communication 8
  - HLPIU (3xx) 13
  - LEPIU (5xx) 13
  - LLPIU (4xx) 13
  - Multifunction controller (MFC and AMC) 27
  - UAC Primary Controller 36
  - UAC Reserve Controller CB (6xx) and EC (8xx) error codes reported by the RCD 45
  - UAC Reserve Controller Director CB (6xx) and EC (8xx) 38
  - UAC Reserve Controller Director MC 41

## H

- hdwconfig test utility 9
- Hiway Bridge
  - alarm connection 17
  - preferred access cable 16
  - switch settings 13

## M

- Main properties for a TDC 3000 channel 2

## P

- Port properties for a TDC 3000 4

## R

- Redundant Port properties for a TDC 3000 channel 5
- reserve controller commands 5

## T

- TDC Data Hiway commands 4

## U

- utility
  - hdwconfig 9







**Honeywell Process Solutions**

2500 West Union Hills Drive

Phoenix AZ 85027

USA

[www.honeywell.com](http://www.honeywell.com)

EP-DSX114

11/05

© 2005 Honeywell International Inc

**Honeywell**