Honeywell

Experion PKS

Honeywell TDC 3000 Data Hiway Interface Reference

EPDOC-XX61-en-431A
February 2015

Release 431

Honeywell

Document	Release	Issue	Date
EPDOC-XX61-en-431A	431	0	February 2015

Disclaimer

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

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

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

Contents

Planning considerations for installing and configuring Honeywell TDC 3000 Data Hiway controllers 7

- Other documentation for Honeywell TDC 3000 Data Hiway 9
- Honeywell TDC 3000 Data Hiway-specific terms 10
- Devices supported by the Honeywell TDC 3000 Data Hiway interface 12
  - Comparing Honeywell TDC 3000 Data Hiway with Experion 13
  - Operational considerations for Honeywell TDC 3000 Data Hiway 15
- Architectures for Honeywell TDC 3000 Data Hiway 16
  - Architectures for Data Hiway Ethernet Bridge (DHEB) 16
  - Architectures for Fault Tolerant Ethernet (FTE) 17
  - Architectures for Data Hiway Bridge (DHB) 17
  - DHEB hardware 18
  - DHEB hardware setup 19
  - DHEB connections 19
  - DHEB grounding considerations 19
  - DHEB software installation 20
  - DHEB software configuration 20
  - DHEB troubleshooting 22
  - DHB hardware 23
  - DHB hardware setup 24
  - Hiway Bridge switch settings 24
  - DHB connections 25
  - DHB grounding considerations 27
  - DHB troubleshooting 29

Honeywell TDC 3000 Data Hiway channel and controller reference 31

- Main properties for a Honeywell TDC 3000 Data Hiway channel 32
- Port properties for a Honeywell TDC 3000 Data Hiway channel 34
- Redundant port properties for a Honeywell TDC 3000 Data Hiway channel 36
- Configuring a Honeywell TDC 3000 Data Hiway channel 37
- Main properties for a Honeywell TDC 3000 Data Hiway controller 38
- I/O slot definitions for Honeywell TDC 3000 Data Hiway 40
- Optimizing Honeywell TDC 3000 Data Hiway scanning performance 41
- Additional tasks for Honeywell TDC 3000 Data Hiway 42
- Commands for Honeywell TDC 300 Data Hiway 43
  - Data Hiway switch command 43
  - Basic Controller (CB) and Extended Controller (EC) commands 43
  - Multifunction Controller (MC) commands 43
  - Reserve Controller Director (RCD) commands 44
  - PIU, DHP, and HIM commands 44
- Sequence programs for Honeywell TDC 3000 Data Hiway 46
- Backing up and restoring Honeywell TDC Data Hiway box configuration 47
- Configuration management for Honeywell TDC 3000 Data Hiway 49

Honeywell TDC 3000 Data Hiway points reference 51

- Configuring exception scanning with Honeywell TDC 3000 Data Hiway 53
- TDC point property settings 55

Defining a Honeywell TDC 3000 Data Hiway address for a point parameter 58
Addressing a Basic Controller (CB) for a Honeywell TDC 3000 Data Hiway 58
Addressing an Extended Controller (EC) for a Honeywell TDC 3000 Data Hiway 59
Addressing a Multifunction Controller (MFC and AMC) for a Honeywell TDC 3000 Data Hiway 60
Addressing a Process Interface Unit (PIU) for a Honeywell TDC 3000 Data Hiway 64
Addressing a Data Hiway Port (DHP) and an LCS620 HIM for a Honeywell TDC 3000 Data Hiway 65
Addressing generic hardware for a Honeywell TDC 3000 Data Hiway 66
Linearization algorithms for Honeywell TDC 3000 Data Hiway 67
Point displays for Honeywell TDC 3000 Data Hiway 68

Control Language reference 71

Configuration tasks 72
CL acronym libraries 72
Process module points 72
C Link addresses 73
Logic blocks 73
Program development 75
CL software environment 75
CL program directories 76
Creating CL programs 76
CL sequence messages 77
Compiling CL programs 77
Compile time errors 78
Downloading sequence programs 78
Troubleshooting sequence programs 79
Sequence program run time errors 79
Uploading sequence programs 79
Comparison with the TPS CL MC compiler 79
Control Language basics 82
Rules and elements 82
Data types 90
Variables and declarations 93
Expressions and conditions 96
Statements 102
Program statement definitions 102
Embedded compiler directives 119
Structures 120
Sequence program definition 120
Abnormal condition handlers 123
Restart routines 125
Accessing parameters 126
User-written subroutines 134
Built-in functions and subroutines 136
Syntax summary 138
Syntax diagram summary 139
Notation used for syntax production rules 146
Production rules 147

Troubleshooting Honeywell TDC 3000 Data Hiway issues 151

Testing Honeywell TDC 3000 Data Hiway communications with the server 153
Monitoring the system status of Honeywell TDC 3000 Data Hiway 154
Configuration monitoring of the Honeywell TDC Data Hiway box 156
Error codes for Honeywell TDC 3000 Data Hiway 160
Hiway addressing/communication error codes 160
HLPIU (3xx), LLPIU (4xx), and LEPIU (5xx) error codes 163

DHP or 620 HIM error codes 165
Basic controller primary (1xx) and reserve (6xx) error codes 167
Extended controller primary (7xx) and reserve (8xx) error codes 169
Multifunction controller (MFC and AMC) error codes 171
UAC primary controller codes 176
UAC Reserve Controller Director CB (6xx) and EC (8xx) error codes 177
UAC Reserve Controller Director MC error codes 178
UAC reserve controller CB (6xx) and EC (8xx) error codes reported by the RCD 181
UAC reserve controller MC error codes reported by the RCD 183

Notices 187

Documentation feedback 188
How to report a security vulnerability 189
Support 190
Training classes 191

Planning considerations for installing and configuring Honeywell TDC 3000 Data Hiway controllers

This reference provides the information you need to set up, configure, and test TDC 3000 Data Hiway controller communications with the server.

Revision history

Revision	Date	Description
A	February 2015	Initial release of document.

How to use this guide

These are the steps for connecting and configuring a TDC 3000 Data Hiway controller. Complete each step before commencing the next.

Steps	Go to
Determine TDC configuration	Architectures for Honeywell TDC 3000 Data Hiway
Use Quick Builder to define channels	• Honeywell TDC 3000 Data Hiway channel and controller reference • "Build channels" topic in the *Quick Builder User's Guide*
Use Quick Builder to define controllers	• Honeywell TDC 3000 Data Hiway channel and controller reference • "Build controllers" topic in the *Quick Builder User's Guide*
Download channel and controller definitions to the server	"Downloading items" topic in the *Quick Builder User's Guide*
Test communications	Testing Honeywell TDC 3000 Data Hiway communications with the server
Troubleshooting communication errors	Troubleshooting Honeywell TDC 3000 Data Hiway issues
Use Quick Builder to define points	Defining a Honeywell TDC 3000 Data Hiway address for a point parameter

Related topics

"Other documentation for Honeywell TDC 3000 Data Hiway" on page 9
"Honeywell TDC 3000 Data Hiway-specific terms" on page 10
"Devices supported by the Honeywell TDC 3000 Data Hiway interface" on page 12
"Architectures for Honeywell TDC 3000 Data Hiway" on page 16
"Honeywell TDC 3000 Data Hiway channel and controller reference" on page 31
"Testing Honeywell TDC 3000 Data Hiway communications with the server" on page 153
"Troubleshooting Honeywell TDC 3000 Data Hiway issues" on page 151

"Defining a Honeywell TDC 3000 Data Hiway address for a point parameter" on page 58

Other documentation for Honeywell TDC 3000 Data Hiway

Experion documentation

- *Software Change Notice* – for the latest enhancements and known problems.
- *Server and Client Planning Guide* – for system overview and planning information.
- *Software Installation User's Guide* – for the system requirements and software installation procedures.
- *Quick Builder User's Guide* – for the configuration of points, controllers, stations.
- *Server and Client Configuration Guide* – for the system configuration procedures.
- *Operator's Guide* – for the system operating procedures.
- Reference guides – for information on specific product options.

TDC 3000 Data Hiway documentation

- 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 documentation

- *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.

Related topics

"Architectures for Honeywell TDC 3000 Data Hiway" on page 16
"Addressing generic hardware for a Honeywell TDC 3000 Data Hiway" on page 66

Honeywell TDC 3000 Data Hiway-specific terms

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.
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.

Devices supported by the Honeywell TDC 3000 Data Hiway interface

Attention

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.
Multifunction Controller	MFC	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.
Advanced Multifunction Controller	AMC	

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	DHP	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 LCS620 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 LCS620 HIM will appear as one to four Data Hiway Ports. The LCS620 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.
LCS620 Hiway Interface Module	HIM	
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	

Related topics

"Comparing Honeywell TDC 3000 Data Hiway with Experion" on page 13

Comparing Honeywell TDC 3000 Data Hiway with Experion

There are many differences between the Honeywell Experion system and the 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 is 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.

Topic	Experion System	TDC 3000 BOS/EOS
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.	4,000 points per BOS, 1,500 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.
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 the titled "CL sequence messages" for more information.	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 Experion Batch Manager (EBM). A simple Batch report is available.	Optional EOS disk set.
TDC group overview display	Use custom displays.	Supported.

Topic	Experion System	TDC 3000 BOS/EOS
TDC Data Hiway commands	The STR command must be used in place of the BOS/EOS INIT command. See the topic titled "Commands for Honeywell TDC 300 Data Hiway" for more information.	Supported.
TDC box types	Supports all box types except for the old Analog Unit box type. Some restrictions apply. See the topic titled "Devices supported by the Honeywell TDC 3000 Data Hiway interface" for more information.	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.

Related topics

"CL sequence messages" on page 77

"Commands for Honeywell TDC 300 Data Hiway" on page 43

"Devices supported by the Honeywell TDC 3000 Data Hiway interface" on page 12

Operational considerations for Honeywell TDC 3000 Data Hiway

This topic identifies some differences that may affect 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. See the topic titled "TDC point property settings" for more information.
- 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. See the topic titled "Configuration management for Honeywell TDC 3000 Data Hiway" for more information about data sources.
- 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.

Related topics

"TDC point property settings" on page 55

"Configuration management for Honeywell TDC 3000 Data Hiway" on page 49

Architectures for Honeywell TDC 3000 Data Hiway

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 topic titled "Other documentation for Honeywell TDC 3000 Data Hiway."

Related topics

“Architectures for Data Hiway Ethernet Bridge (DHEB)” on page 16
“Architectures for Fault Tolerant Ethernet (FTE)” on page 17
“Architectures for Data Hiway Bridge (DHB)” on page 17
“DHEB hardware” on page 18
“DHEB hardware setup” on page 19
“DHEB connections” on page 19
“DHEB grounding considerations” on page 19
“DHEB software installation” on page 20
“DHEB software configuration” on page 20
“DHEB troubleshooting” on page 22
“DHB hardware” on page 23
“DHB hardware setup” on page 24
“Hiway Bridge switch settings” on page 24
“DHB connections” on page 25
“DHB grounding considerations” on page 27
“DHB troubleshooting” on page 29
“Other documentation for Honeywell TDC 3000 Data Hiway” on page 9

Architectures for Data Hiway Ethernet Bridge (DHEB)

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.

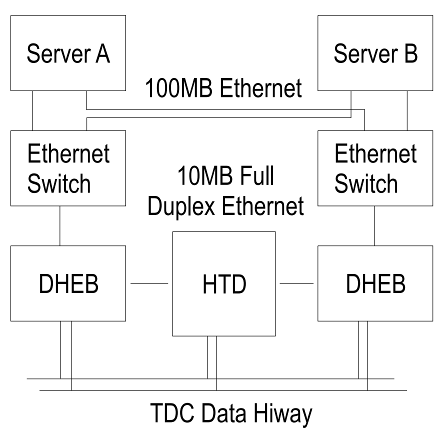


Figure 1: Fully redundant single Data Hiway system using two Data Hiway Ethernet Bridges

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.

Attention
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.

Architectures for Fault Tolerant Ethernet (FTE)

The DHEB does not support dual Ethernet connections but it can be connected to a 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*.

Architectures for Data Hiway Bridge (DHB)

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.

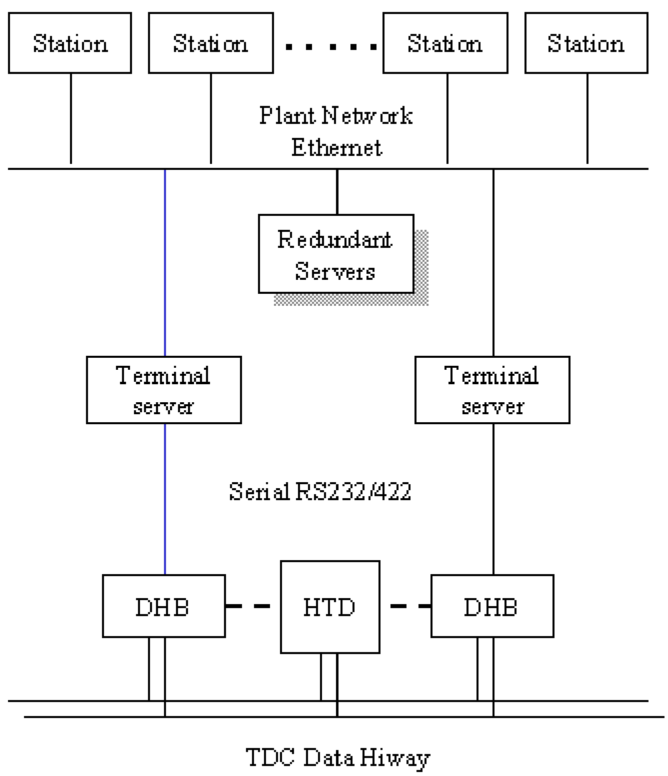


Figure 2: Fully redundant single Data Hiway system using two Data Hiway Bridges

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 backup path automatically. During normal operations, the server will perform a diagnostic scan via the backup path every 60 seconds.

See the *Software Installation User's 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.

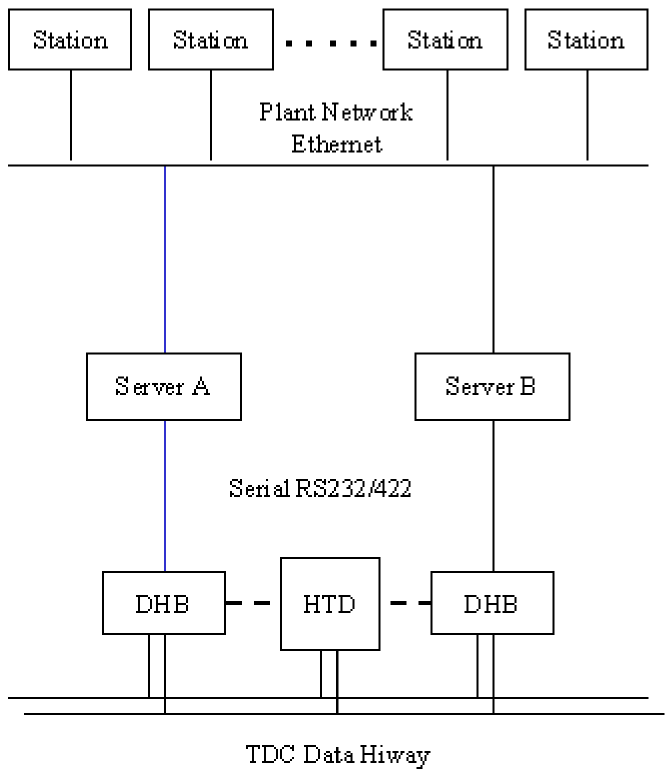


Figure 3: Alternative redundant single Data Hiway system without terminal servers using two Data Hiway Bridges

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 failover must be performed in order to reestablish communications with the TDC Data Hiway.

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.

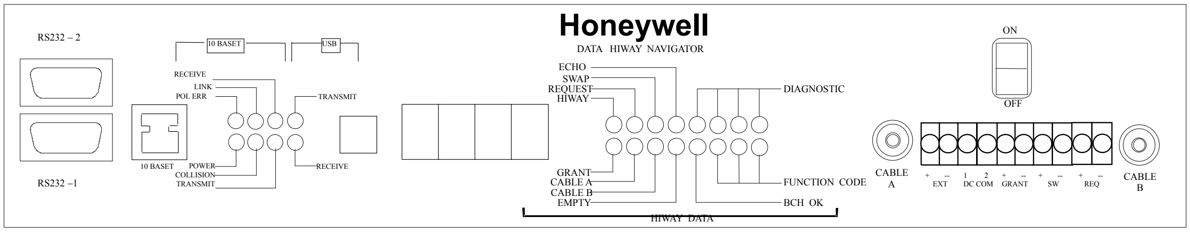


Figure 4: DHEB front panel

The rear panel contains the 24V DC, AC, and Ground connections. The AC input range is 85–264 volts.

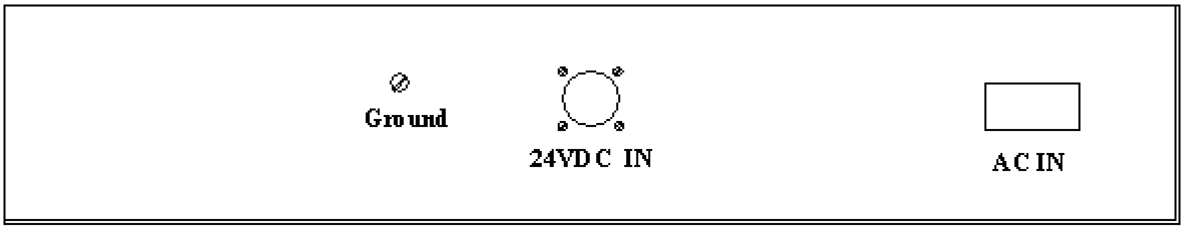


Figure 5: DHEB rear panel

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 DHEB 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 DHEB 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* for DHEB connection information.

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.

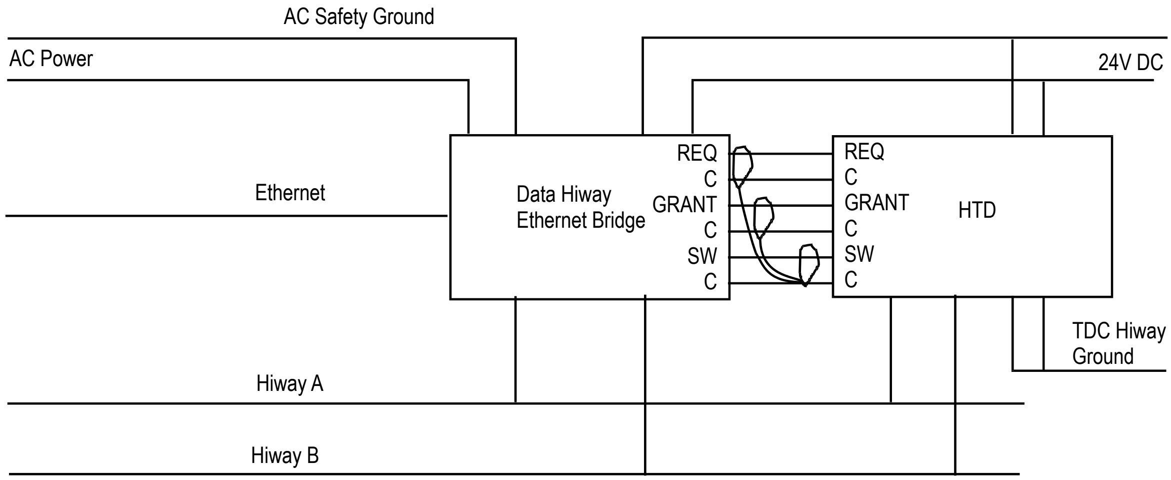


Figure 6: DHEB grounding considerations

The 24V DC 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, restart the computer to complete the installation.
6. For redundant Experion systems, repeat the software installation on the other server.

DHEB software configuration

CAUTION

A DHEB can handle only four connections at one time. Experion uses three of these connections, plus another one when you use the test harness. Therefore, keeping the **DHEB Monitor** program running could affect Experion performance. Close the **DHEB Monitor** program as soon as you have finished using it.

After completing the software installation, start the DHEB Monitor program and configure the interface as follows:

1. Select **Start** > **All Programs** > **DHEB** > **DHEB MONITOR** to start the DHEB Monitor program.
2. Select **Configure** > **DHEB Interface**. The **Interface Configuration** dialog box opens.
3. Click **Edit** and enter the configuration details, where:
   - **Hiway Number** corresponds with the DHEB switch settings, see the DHEB setup.
   - **Hiway Description** with the recommended entry as *HIWAYnn* where *nn* is the same as the Hiway Number.
   - **IP Address** is the address assigned to the DHEB, see the DHEB setup procedure.

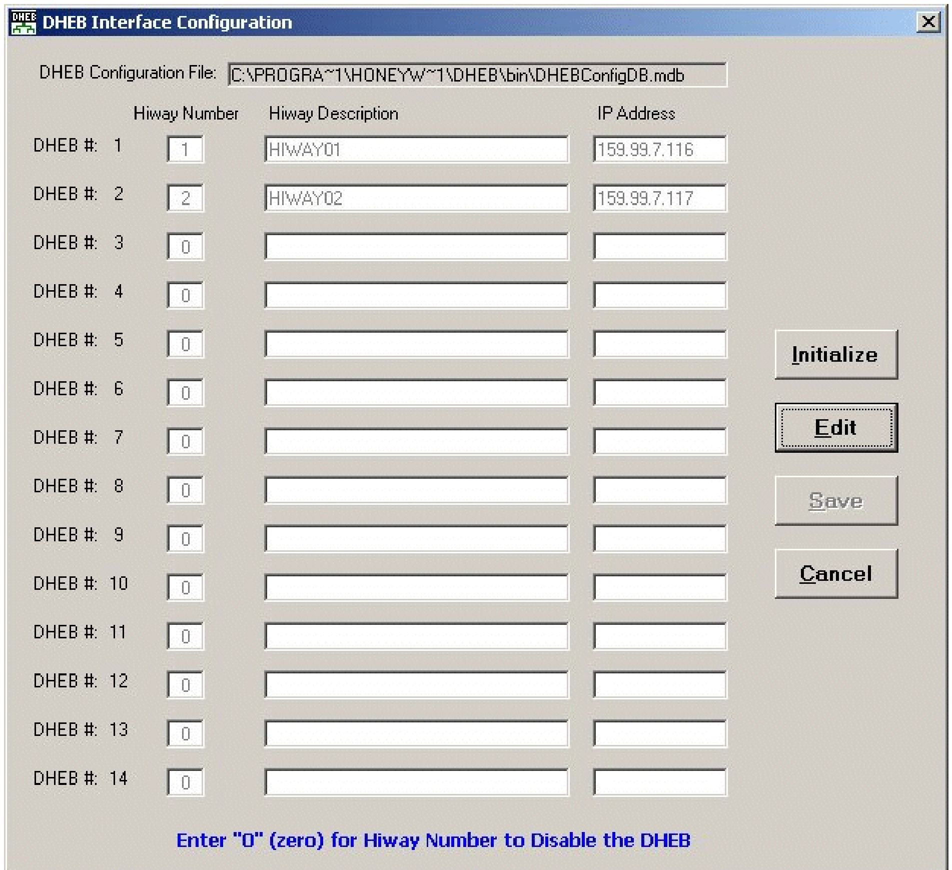


Figure 7: DHEB Interface Configuration

4 Click **Save** to save the DHEB interface configuration.

The following error appears if the Monitor program cannot communicate with the DHEB.

Figure 8: DHEB Monitor Error

The following dialog box is displayed if the communications are successful.

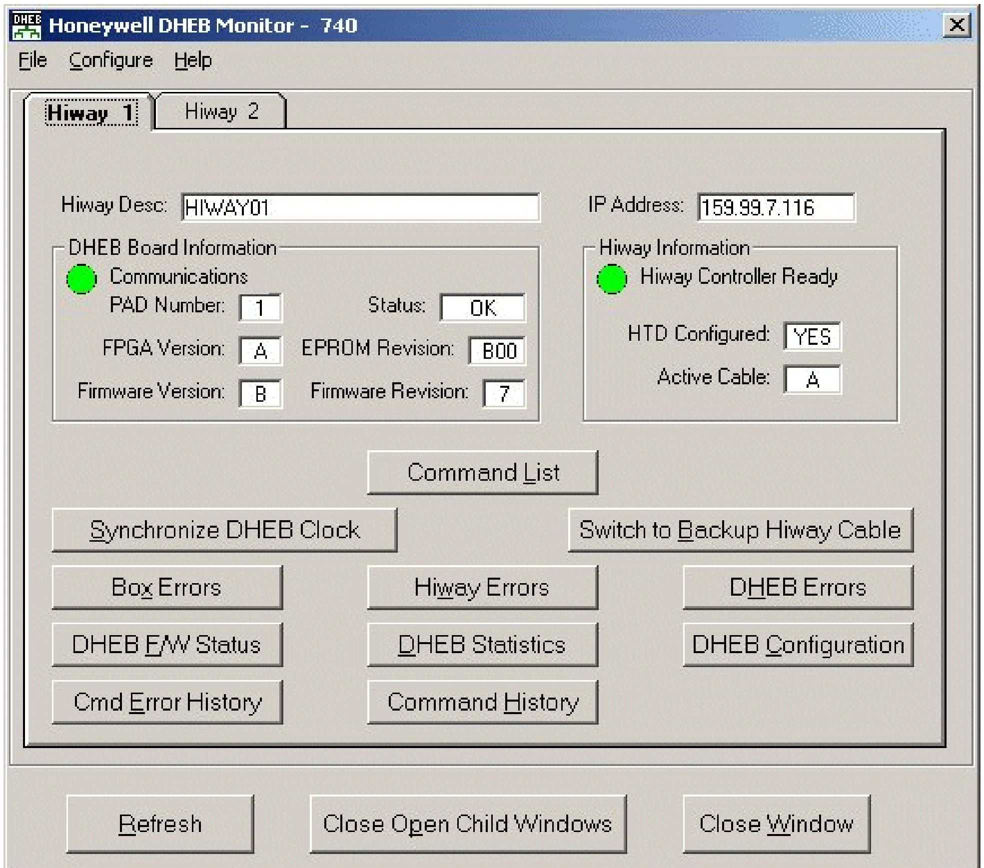


Figure 9: Honeywell DHEB Monitor

If this dialog box does not appear, see the topic titled "DHEB troubleshooting" for more information.

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. Because of this, Honeywell recommends that you do not use the **DHEB Monitor** program to read/write to the online process connected boxes.

Results

If the communications are successful, go to the Experion system configuration procedures. See the topic titled "Honeywell TDC 3000 Data Hiway channel and controller reference" for more information.

Related topics

“DHEB troubleshooting” on page 22
“Honeywell TDC 3000 Data Hiway channel and controller reference” on page 31
“Configuration management for Honeywell TDC 3000 Data Hiway” on page 49
“Port properties for a Honeywell TDC 3000 Data Hiway channel” on page 34

DHEB troubleshooting

If communications are not successful, follow these steps:

1. Check the DHEB front panel Hexadecimal 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** > **All 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 six 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 DHEB 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** > **All Programs** > **Honeywell Experion PKS** > **Server** > **Diagnostic Tools** > **Experion PKS Server Log**.

4 If you have correctly configured the DHEB and you get an error message "The requested hiway is not configured," check the Windows advanced security settings of the DHEB configuration folder, *C:\ProgramData\Honeywell\DHEB*, and the file contained within it. The folder and the file should inherit security permissions from the parent object, allowing access to the folder and file by all Honeywell groups.

Related topics

"DHEB software configuration" on page 20

DHB hardware

The DHB is designed for installation in a 19 inch equipment rack.

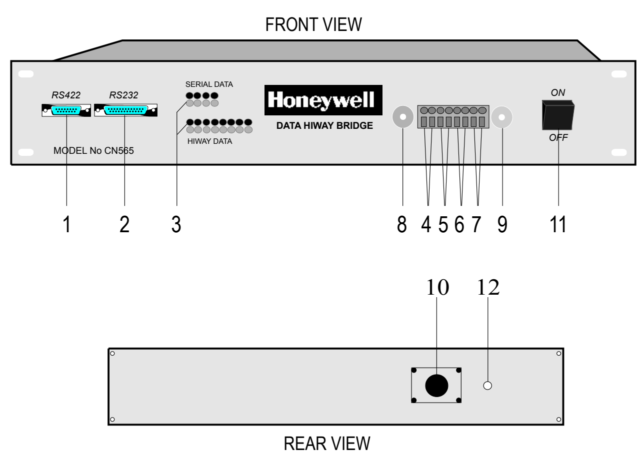


Figure 10: DHB front and rear views

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 (baud should be fast for best performance, 19,200 or higher)
- Parity (usually EVEN)
- Protocol (RS-232)
- Hiway Address (3, for example)

Record these settings because you will need them when you configure the channel on the server.

Hiway Bridge switch settings

Switch bank S3

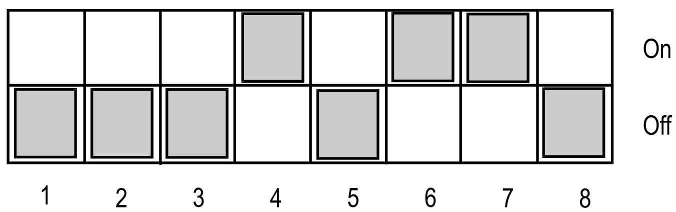


Figure 11: Settings for S3

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 as shown in the following table.

Switch 8 is ON if Hiway writes are to be inhibited (the default is OFF).

Attention
The following settings depend on the IC used in position U6 on the processor board BM403. The default baud is 9600.

SW4 to SW7[1]	Baud if U6 is 68661PB or SCN26661BC	Baud 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
1110	600	1200
0001	1200	1800
1001	1800	2000
0101	2000	2400

[1] **0 = OFF, 1 = ON**

SW4 to SW7[1]	Baud if U6 is 68661PB or SCN26661BC	Baud if U6 is 68661PC
1101	2400	3600
0011	4800	4800
1011	9600	7200
0111	19,200	9600
1111	38,400	19,200

Switch bank S2

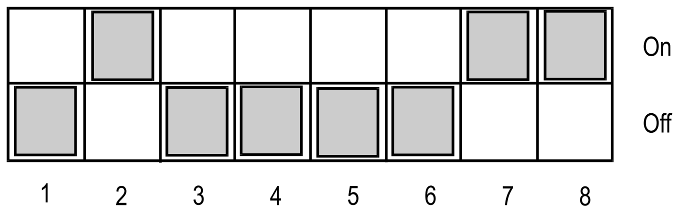


Figure 12: Settings for S2

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[2]	Data Hiway Address
100000	1
010000	2
110000	3

Attention

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

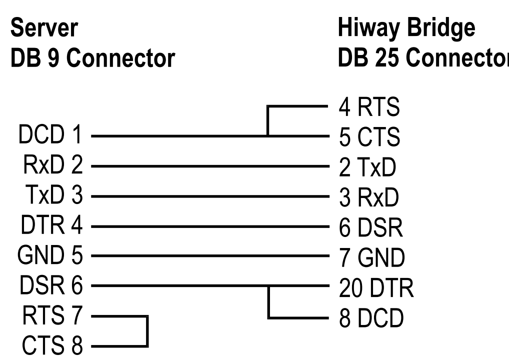


Figure 13: RS-232 Wiring for server with DB 9 connector

[1] **0 = OFF, 1 = ON**

[2] **0 = OFF, 1 = ON**

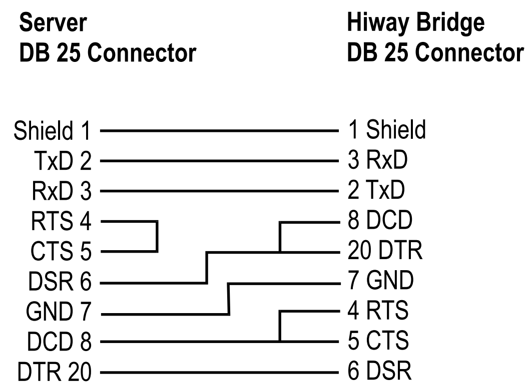


Figure 14: RS-232 Wiring for server with DB 25 connector

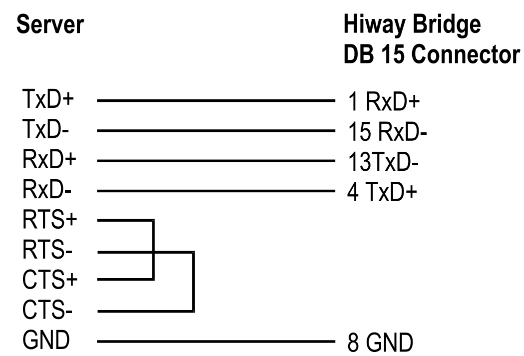


Figure 15: RS-422 wiring

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 24V DC power supply as follows.

Wire	PSU
Red	24V
Black	0V, common

Attention

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.

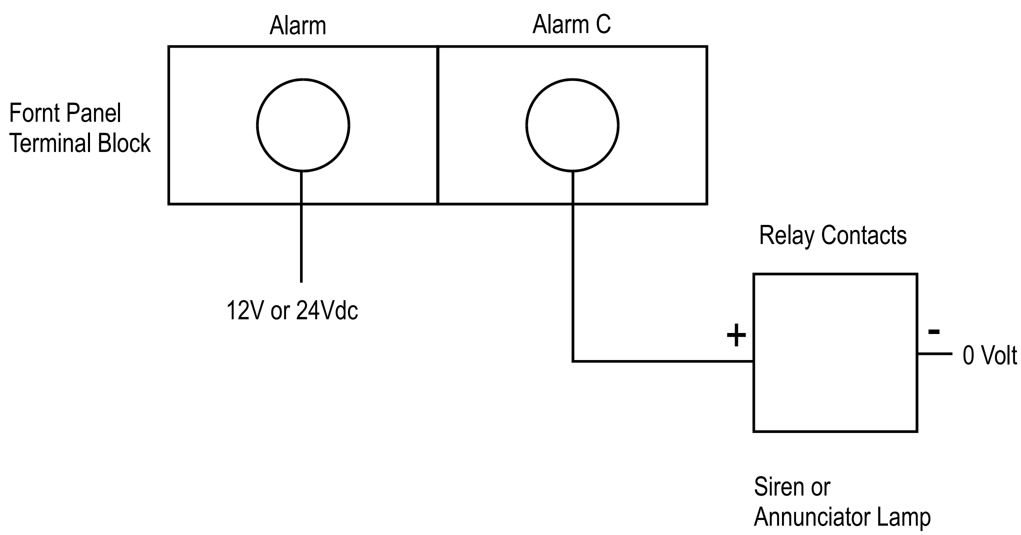


Figure 16: Recommended External Connection To Alarms

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.

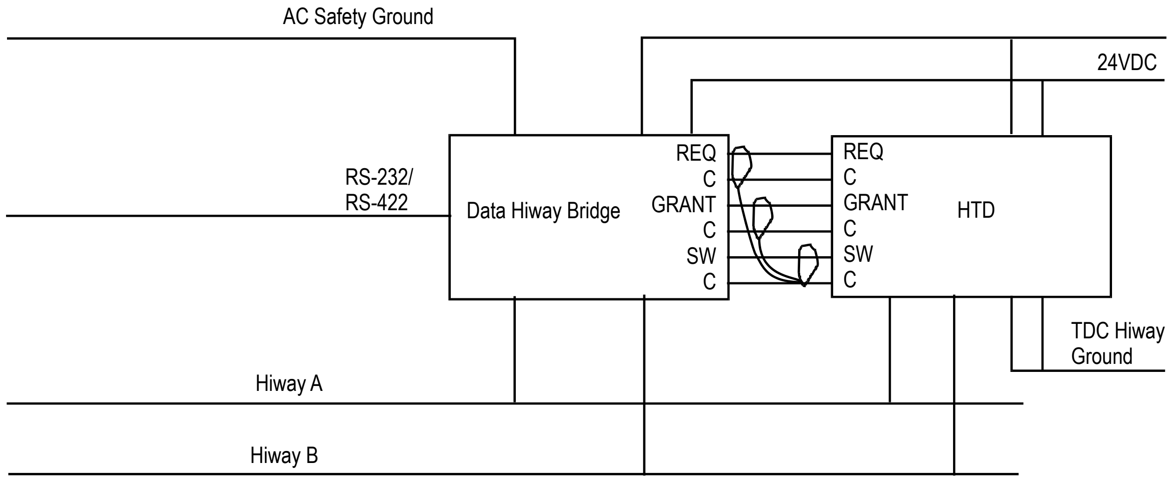


Figure 17: DHB grounding only at the HTD

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.

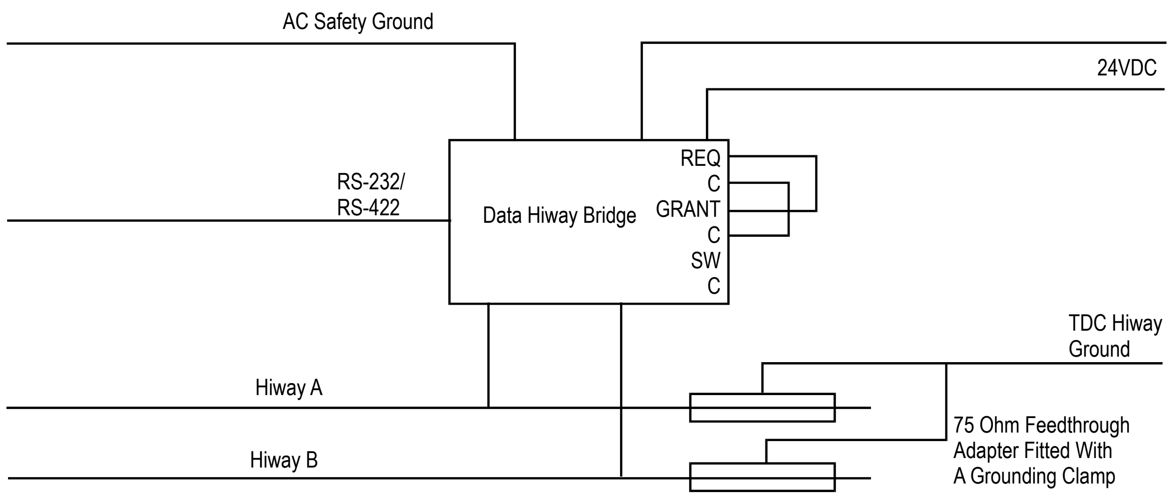


Figure 18: DHB grounding only at the Data Hiway Bridge

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 24V DC 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 completed before the communications can be tested. See the topic titled "Honeywell TDC 3000 Data Hiway channel and controller reference" for more information.

Related topics

“Honeywell TDC 3000 Data Hiway channel and controller reference” on page 31

Honeywell TDC 3000 Data Hiway channel and controller reference

This section describes the configuration and addressing information specific to Honeywell TDC 3000 Data Hiway channels and controllers.

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

Related topics

"Main properties for a Honeywell TDC 3000 Data Hiway channel" on page 32
"Port properties for a Honeywell TDC 3000 Data Hiway channel" on page 34
"Redundant port properties for a Honeywell TDC 3000 Data Hiway channel" on page 36
"Configuring a Honeywell TDC 3000 Data Hiway channel" on page 37
"Main properties for a Honeywell TDC 3000 Data Hiway controller" on page 38
"I/O slot definitions for Honeywell TDC 3000 Data Hiway" on page 40
"Optimizing Honeywell TDC 3000 Data Hiway scanning performance" on page 41
"Additional tasks for Honeywell TDC 3000 Data Hiway" on page 42
"Commands for Honeywell TDC 300 Data Hiway" on page 43
"Sequence programs for Honeywell TDC 3000 Data Hiway" on page 46
"Backing up and restoring Honeywell TDC Data Hiway box configuration" on page 47
"Configuration management for Honeywell TDC 3000 Data Hiway" on page 49
"Planning considerations for installing and configuring Honeywell TDC 3000 Data Hiway controllers" on page 7
"DHB troubleshooting" on page 29
"DHEB software configuration" on page 20
"Configuration management for Honeywell TDC 3000 Data Hiway" on page 49

Main properties for a Honeywell TDC 3000 Data Hiway channel

The Main tab defines the basic properties for a Honeywell TDC 3000 Data Hiway channel.

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

Property	Description
Name	The unique name of the channel. A maximum of 10 alphanumeric characters (no spaces or double quotes). Note: In Station displays, underscore characters (_) appear as spaces.
Description	(Optional) A description of the channel. A maximum of 132 alphanumeric characters, including spaces.
Marginal Alarm Limit	The communications alarm marginal limit at which the channel is declared to be marginal. When this limit is reached, a high priority alarm is generated. To change the priority of the alarm system wide, see the topic titled "Configuring system alarm priorities" in the *Server and Client Configuration Guide*. To change the priority of the alarm for one channel, see the topic titled "About configuring custom system alarm priorities for an individual channel or controller" in the *Server and Client Configuration Guide*. A channel barometer monitors the total number of requests and the number of times the controller did not respond or response was incorrect. The barometer increments by two or more, depending on the error, and decrements for each good call. To calculate an acceptable marginal alarm limit, use the formula: Square root of the number of controllers on the channel × Marginal Alarm Limit defined on those controllers (Normally, you specify the same value for all controllers on a channel). For example, if there are 9 controllers on the channel and their Marginal Alarm Limit is set to 25, the value would be 3 (which is the square root of 9) x 25 = 75.
Fail Alarm Limit	The communications alarm fail limit at which the channel is declared to have failed. When this barometer limit is reached, an urgent alarm is generated. To change the priority of the alarm system wide, see the topic titled "Configuring system alarm priorities" in the *Server and Client Configuration Guide*. To change the priority of the alarm for one channel, see the topic titled "About configuring custom system alarm priorities for an individual channel or controller" in the *Server and Client Configuration Guide*. Set this to double the value specified for the channel Marginal Alarm Limit.
Connect Timeout	The length of time that the server attempts to connect to the controller. The server will stop trying to connect to the controller once the timeout period passes. The default value 10 seconds. Use the default value unless the communications line has a high error rate, or unless you are using modems.
Read Timeout	The length of time that the server will wait for a reply from the controller. The server will stop waiting once the timeout period passes. The default value is 2 seconds. Use the default value unless the communications line has a high error rate, or unless you are using modems.
Item Type	The type of channel specified when this item was created.
Last Modified	The date and time the channel properties were modified.
Last Downloaded	The date and time the channel was last downloaded to the server.

Property	Description
Item Number	The unique item number currently assigned to this channel, in the format *CHNcc*, where *cc* is the channel number. You can change the item number if you need to match your current server database configuration. The number must be between *01* and the maximum number of channels allowed for your system. For more information about setting the maximum value, see the topic titled "Adjusting sizing of non-licensed items" in the *Supplementary Installation Tasks Guide*.

Port properties for a Honeywell TDC 3000 Data Hiway channel

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:

- *Serial* for DHB Channel with a direct connection to the server. See the section below titled "Serial port properties."
- *Terminal Server* for DHB Channel with a terminal server connection to the server. See the section below titled "Terminal server port properties."
- *LANVendor* for a DHEB Channel. See the section below titled "LANVendor port properties."

Serial port properties

Attention
For Redundant Channels, the Port and Redundant Port settings are normally identical

Property	Description
Serial Port Name	The device name of the serial port.
Baud	The number of data bits per second. The default is *9600*.
Number of Data Bits	The number of data bits used for transmission. The default is *8*.
Stop Bits	The number of stop bits used for transmission The default is *1*.
Parity	Defines parity verification of each character and must match configuration on the end device. The default is *NONE*.
Checksum	The type of checksum error detection used for the port. 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	Not applicable, leave as *NONE*.
RS-232 settings	Not applicable.
RS-485 settings	Not applicable.

Terminal server port properties

Attention
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 Terminal Server TCP Port No	The name and port number of terminal server to which the channel is connected. You can specify either a TCP host name or an IP address, but it must match the TCP host name used when you installed and internally configured the terminal server.

Property	Description
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, leave as *NONE*.

LANVendor port properties

Attention
For Redundant Channels, the Port Names are different for example, *HIWAY01* and *HIWAY02*.

Property	Description
Port Name	This entry must have the format *HIWAYnn* where *nn* is the Hiway Number. See the topic titled "DHEB software configuration" for more information.

Related topics

"DHEB software configuration" on page 20
"Redundant port properties for a Honeywell TDC 3000 Data Hiway channel" on page 36

Redundant port properties for a Honeywell TDC 3000 Data Hiway channel

The Redundant Port tab defines the communication-related properties for a redundant channel, which can be either:

- *Serial*
- *TerminalServer*
- *LANVendor*

See the topic titled "Port properties for a Honeywell TDC 3000 Data Hiway channel" for more information.

Related topics

"Port properties for a Honeywell TDC 3000 Data Hiway channel" on page 34

Configuring a Honeywell TDC 3000 Data Hiway channel

Once the channel is downloaded to the server, you can use Station to configure certain channel properties.

Prerequisites

- You are signed-in to Station with *mngr* security level.

To configure TDC channels

1. In Station, select **Configure** > **System Hardware** > **SCADA Controllers**.
   The **SCADA Controllers** page appears.
2. In the **Channel** column, click the TDC channel to configure.
   Details for the channel appear on two tabs: **Status Details** tab and **Configuration** tab.
3. On the **Configuration** tab, configure as follows:

Property	Description
Upload permitted	When selected, the TDC Box configuration can be uploaded (saved) to the server.
Download permitted	When selected, the TDC Box configuration can be downloaded (restored) from the server.
Refresh enabled for computer mode timeout	When 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	When 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 the topic titled "Configuring exception scanning with Honeywell TDC 3000 Data Hiway" for more information.
PIU/DHP significant change reporting	When 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. You must also select the **Hiway Owner** check box for this function to work. See the topic titled "Configuring exception scanning with Honeywell TDC 3000 Data Hiway" for more information.

Related topics

“Configuration management for Honeywell TDC 3000 Data Hiway” on page 49
“Configuring exception scanning with Honeywell TDC 3000 Data Hiway” on page 53

Main properties for a Honeywell TDC 3000 Data Hiway controller

The **Main** tab defines the basic properties for a Honeywell TDC 3000 Data Hiway controller.

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

Property	Description
Name	A unique name for example, *AMChhbb* where *hh* is the Hiway Number/Channel Number and *bb* is the Controller Number/Box Address, using a maximum of *10* alphanumeric characters (no spaces, underscores, or double quotes).
Description	(Optional) A description of the controller. A maximum of *132* alphanumeric characters, including spaces.
Associated Asset	The Tag Name of the Asset to be associated with the alarm group.
Channel Name	The name of the channel on which the controller communicates with the server. (You must have already defined a channel for it to appear in this list.)
Marginal Alarm Limit	The communications alarm marginal limit at which the controller is declared to be marginal. When this limit is reached, a high priority alarm is generated. To change the priority of the alarm system wide, see the topic titled "Configuring system alarm priorities" in the *Server and Client Configuration Guide*. To change the priority of the alarm for one controller, see the topic titled "About configuring custom system alarm priorities for an individual channel or controller" in the *Server and Client Configuration Guide*. A controller barometer monitors the total number of requests and the number of times the controller did not respond or response was incorrect. The barometer increments by two or more, depending on the error, and decrements for each good call. The default value is *25*.
Fail Alarm Limit	The communications alarm fail limit at which the controller is declared to have failed. When this barometer limit is reached, an urgent alarm is generated. To change the priority of the alarm system wide, see the topic titled "Configuring system alarm priorities" in the *Server and Client Configuration Guide*. To change the priority of the alarm for one controller, see the topic titled "About configuring custom system alarm priorities for an individual channel or controller" in the *Server and Client Configuration Guide*. Set this to double the value specified for the controller Marginal Alarm Limit. The default is *50*.
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 Type	The type of controller specified when this item was created.
Last Modified	The date and time the controller properties were modified.
Last Downloaded	The date and time the controller was last downloaded to the server.

Property	Description
Item Number	The unique item number currently assigned to this controller, in the format `RTUnnnnn`. You can change the item number if you need to match your current server database configuration. The number must be between `01` and the maximum number of controllers allowed for your system. For more information about setting the maximum value, see the topic titled "Adjusting sizing of non-licensed items" in the *Supplementary Installation Tasks Guide*.

Related topics

"CL acronym libraries" on page 72

I/O slot definitions for Honeywell TDC 3000 Data Hiway

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 offline 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 **mngr**.
3. Select **Configure** > **System Hardware** > **SCADA Controllers** to call up the SCADA Controllers 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 topic titled "Configuration monitoring of the Honeywell TDC Data Hiway box" for more 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 offline 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 offline 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 topic titled "Backing up and restoring Honeywell TDC Data Hiway box configuration" for more information.

Attention

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.

Related topics

“Configuration management for Honeywell TDC 3000 Data Hiway” on page 49
“Configuration monitoring of the Honeywell TDC Data Hiway box” on page 156
“Backing up and restoring Honeywell TDC Data Hiway box configuration” on page 47

Optimizing Honeywell TDC 3000 Data Hiway scanning performance

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 five 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.

Related topics

"TDC point property settings" on page 55

Additional tasks for Honeywell TDC 3000 Data Hiway

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 the section titled "Control Language reference" for more information.

See the *Server and Client Configuration Guide* for all other tasks, for example, reports.

Related topics

"Control Language reference" on page 71
"Configuration management for Honeywell TDC 3000 Data Hiway" on page 49

Commands for Honeywell TDC 300 Data Hiway

The Data Hiway commands are used to switch to the backup Data Hiway cable and to control the state of the Data Hiway boxes.

Attention

The Data Hiway commands use the Controller Number as displayed on the SCADA Controllers 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" displays when the command is successfully delivered to the Data Hiway box. After entering a command, the status of the Data Hiway box should checked on the SCADA Controllers displays.

Attention

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.

Related topics

"Data Hiway switch command" on page 43
"Basic Controller (CB) and Extended Controller (EC) commands" on page 43
"Multifunction Controller (MC) commands" on page 43
"Reserve Controller Director (RCD) commands" on page 44
"PIU, DHP, and HIM commands" on page 44
"Comparing Honeywell TDC 3000 Data Hiway with Experion" on page 13

Data Hiway switch command

Attention

The HTD can switch cables only once, the switch back must be performed via HTD panel.

Command	Description
SWT nn	Request DHB or HTD to switch to the backup Data Hiway cable.

Basic Controller (CB) and Extended Controller (EC) commands

Command	Description
STR nn	Restart controller *nn* and initialize the box address locations in the controller.

Multifunction Controller (MC) commands

Command	Description
STR nn	Restart controller *nn* and initialize the box address locations in the controller.
PRC nn	Change the state of controller *nn* to PROCESSING.
IDL nn	Change the state of controller *nn* to IDLE.

Reserve Controller Director (RCD) commands

Command	Description
`STR dd`	Restart Reserve Controller Director *dd* and initialize the box address locations in the controller.
`STR dd 00`	Restart the Reserve Controller associated with Reserve Controller Director *dd*.
`STR dd nn`	Restart Primary Controller nn through Reserve Controller Director *dd*.
`TST nn`	Test backup (switch) from Primary Controller *nn* to its associated Reserve Controller.
`SAV nn`	Back up (switch) from Primary Controller *nn* to its associated Reserve Controller.
`ENB nn`	Enable back up of Primary Controller *nn*.
`DIS nn`	Disable back up of Primary Controller *nn*.
`SWB dd`	Switch back from Reserve to the Primary Controller through the Reserve Controller Director *dd*.

PIU, DHP, and HIM commands

Command	Description
`CLR nn`	Enable processing for controller *nn* and initialize the box address locations in the controller.
`STR nn`	This command clears the controller memory and restarts controller *nn*. 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.

Sequence programs for Honeywell TDC 3000 Data Hiway

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 SCADA Controllers 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.

Attention

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 topic titled "CL sequence messages" for more information.

Related topics

“CL sequence messages” on page 77

Backing up and restoring Honeywell TDC Data Hiway box configuration

You use the configuration utility **hdwconfig** to back up 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.

Attention

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:

```
HDWCONFIG RESTORE file {-CHN cc| cc-RTU rr} [-FROM nn]
```

Attention

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.

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 file	Saves the configuration image to *file*.
RESTORE file	Restores the configuration from *file*.
LIST file	Displays a list of the channels and controllers contained in *file*.
-CHN cc	Channel *cc*
-RTU rr	Controller *rr*
-RENEW	Forces data for the controller to be saved, even if data already exists in the file.
-FROM nn	Used in conjunction with *-RTU nn* to restore controllers from *nn* to *rr*.

Examples

To save controller 21's configuration as file *box21*, type:

```
hdwconfig save box21 -rtu 21
```

To save the configuration for all controllers connected to channel 01 to the file *hiway01*, type:

```
hdwconfig save hiway01 -chn 01
```

To restore the configuration saved as file *box21* to the server image for box 21, type:

```
hdwconfig restore box21 -rtu 21
```

To restore the configuration saved as file *box21* to the server image for box 22, type:

```
hdwconfig restore box21 -rtu 22 -from 21
```

Related topics

"Configuration management for Honeywell TDC 3000 Data Hiway" on page 49
"I/O slot definitions for Honeywell TDC 3000 Data Hiway" on page 40

Configuration management for Honeywell TDC 3000 Data Hiway

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"
Channel definitions	Primary server	Backup server *.qdb* file or *.hdw* file	Use Quick Builder or *.hdw* file as a backup.	"Honeywell TDC 3000 Data Hiway channel and controller reference"
Channel configuration	Primary server	Backup server	**CAUTION** If the channel is rebuilt, the channel configuration information is deleted.	"Configuring a Honeywell TDC 3000 Data Hiway channel"
Controller definitions	Primary server	Backup server *.qdb* file or *.hdw* file	Use Quick Builder or *.hdw* file as a backup.	"Honeywell TDC 3000 Data Hiway channel and controller reference"
Asset definitions	Primary server	Backup server		*Enterprise Model Builder User's Guide*
Asset assignment	Primary server	Backup server	**CAUTION** If a Station is rebuilt, asset assignments are deleted.	*Enterprise Model Builder User's Guide*
TDC controller I/O slot definitions	TDC controller data	Primary and backup server Backup files	Use the **hdwconfig** utility to create backup files.	"I/O slot definitions for Honeywell TDC 3000 Data Hiway" "Backing up and restoring Honeywell TDC Data Hiway box configuration"
Point configuration	Primary server	Backup server *.qdb* file or *.pnt* file	**CAUTION** See point 1 in the Caution notice at the start of this topic.	"Honeywell TDC 3000 Data Hiway points reference"

Configuration Entities	Master Data	Backup Data	Configuration Management	References
TDC controller configuration	TDC controller data	Primary and backup server Backup files	Use the **hdwconfig** utility to create backup files.	"Additional tasks for Honeywell TDC 3000 Data Hiway" "Backing up and restoring Honeywell TDC Data Hiway box configuration"
Group and trend display configuration	Primary server	Backup server `.pnt` file	See the section below titled "Group and Trend display configuration."	
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 contains *only* the group and trend entries. The point build file can then be used to restore the group and trend information if required.

Related topics

"Operational considerations for Honeywell TDC 3000 Data Hiway" on page 15
"DHEB software configuration" on page 20
"Honeywell TDC 3000 Data Hiway channel and controller reference" on page 31
"Configuring a Honeywell TDC 3000 Data Hiway channel" on page 37
"I/O slot definitions for Honeywell TDC 3000 Data Hiway" on page 40
"Backing up and restoring Honeywell TDC Data Hiway box configuration" on page 47
"Honeywell TDC 3000 Data Hiway points reference" on page 51
"Additional tasks for Honeywell TDC 3000 Data Hiway" on page 42
"TDC point property settings" on page 55

Honeywell TDC 3000 Data Hiway points reference

This section describes how to configure points for a Honeywell TDC 3000 Data Hiway controller using Quick Builder.

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

Scanning is the process by which the server requests data from the controllers. The TDC controllers support periodic scanning and exception scanning.

Attention
Both types of scanning can be used with a TDC controller.

Periodic 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.

Exception scanning

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.

CAUTION
The small scan packets generated by exception scanning can severely limit the data throughput.

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 the topic titled "Configuring exception scanning with Honeywell TDC 3000 Data Hiway" for more information.
- 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.

Related topics

"Configuring exception scanning with Honeywell TDC 3000 Data Hiway" on page 53
"TDC point property settings" on page 55
"Defining a Honeywell TDC 3000 Data Hiway address for a point parameter" on page 58
"Configuration management for Honeywell TDC 3000 Data Hiway" on page 49

"Configuring exception scanning with Honeywell TDC 3000 Data Hiway" on page 53

Configuring exception scanning with Honeywell TDC 3000 Data Hiway

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 SCADA Controllers 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.

Attention

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
		Analog significant change (insure every analog subslot significant change is configured)	Yes[3]	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[3]	No
		Counter rollover	Yes	No
	Change flag	SP, OP, MD changes by other PADS	Yes	No

3 If report significant changes enabled.

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.
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 SCADA Controllers display on Station to view and change these values. If a value is required in a point record, use exception scanning.
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.

Related topics

"Configuring a Honeywell TDC 3000 Data Hiway channel" on page 37
"Honeywell TDC 3000 Data Hiway points reference" on page 51

TDC point property settings

For an introduction to assets, see the section titled "Enterprise models" in the *Server and Client Planning Guide*. For details on how to configure assets before building the points, see the *Enterprise Model Builder User's Guide*.

Attention

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 topic titled "I/O slot definitions for Honeywell TDC 3000 Data Hiway" for more information.

Tab	Property	Notes
Main	Point ID	Must be unique, can contain up to 40 alphanumeric characters, with at least one alpha character. Normally prefixed 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`.
	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 the topic titled "Defining a Honeywell TDC 3000 Data Hiway address for a point parameter." 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 the topic titled "Optimizing Honeywell TDC 3000 Data Hiway scanning performance."
	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. See the topic titled "Configuration management for Honeywell TDC 3000 Data Hiway."
	Point Detail Display	Special point detail display templates are available for TDC Extended Controller analog points. See the topic titled "Point displays for Honeywell TDC 3000 Data Hiway."

Tab	Property	Notes
	Group Faceplate Template Display	A special group display templates is available for TDC analog points. See the topic titled "Point displays for Honeywell TDC 3000 Data Hiway."
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 the topic titled "Defining a Honeywell TDC 3000 Data Hiway address for a point parameter." 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.
	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 the topic titled "Defining a Honeywell TDC 3000 Data Hiway address for a point parameter." 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 the topic titled "Optimizing Honeywell TDC 3000 Data Hiway scanning performance."
	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%.

Tab	Property	Notes
	Analog Point - Set point Limits	These limits are independent of the TDC box set point limits. To use the TDC box set point limits, disable the points set point limits by setting the values outside of the configured range in engineering units.
	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`.
	Status Point – Output State/Target Input State Associations	See the *Quick Builder User's Guide* 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 *Server and Client Configuration Guide* for details.
User Defined		Used to add user defined parameters to the point, see the *Quick Builder User's Guide* for details.

Related topics

"Operational considerations for Honeywell TDC 3000 Data Hiway" on page 15
"Defining a Honeywell TDC 3000 Data Hiway address for a point parameter" on page 58
"Optimizing Honeywell TDC 3000 Data Hiway scanning performance" on page 41
"Configuration management for Honeywell TDC 3000 Data Hiway" on page 49
"Point displays for Honeywell TDC 3000 Data Hiway" on page 68

Defining a Honeywell TDC 3000 Data Hiway address for a point parameter

The point parameter addresses consist of a controller name as well as an address 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 ....

Attention
Points cannot be built for the TDC Reserve Controller Directors (Basic, Multifunction, and Extended).

Related topics

"Planning considerations for installing and configuring Honeywell TDC 3000 Data Hiway controllers" on page 7
"TDC point property settings" on page 55

Addressing a Basic Controller (CB) for a Honeywell TDC 3000 Data Hiway

The Basic Controller consists of eight modulating slots. The address format is:

AS

where:

Part	Description
A	Is the address acronym listed in "Table 1: Basic Controller addresses".
S	Is the slot number. See "Table 1: Basic Controller addresses" for the valid range.

Examples

- *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)	1–8
Set point	SP	1–8
Output	OP	1–8
Mode	MD	1–8
Gain	K (with a space)	1–8
Computer set point	CS	1–8
Computer output	CO	1–8
Digital filter	TD	1–8
Integral time	T1	1–8
Derivative time	T2	1–8
PV HI limit	PH	1–8

Address Parameter Description	Address Acronym (A)	Slot Range (S)
PV LO limit	PL	1–8
Deviation HI limit	DH	1–8
Deviation LO limit	DL	1–8
Remote variable	RV (read only)	1–8
Bias	BI	1–8
Ratio	RA	1–8

Addressing an Extended Controller (EC) for a Honeywell TDC 3000 Data Hiway

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
A	Is the address acronym listed in “Table 2: Extended Controller addresses”.
S	Is the slot number. See “Table 2: Extended Controller addresses” 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)	1–16
Digital Output (A)	Digital output A	SOA	1–8
Digital Output (B)	Digital output B	SOB	1–8
Analog point			
Modulating	Process variable	PV (read only)	1–16
	Set point	SP	1–16
	Output	OP	1–16
	Mode	MD	1–16
	Gain	K (with a space)	1–16
	Integral time	TI	1–16
	Derivative time	T2	1–16
	Digital filter	TD	1–16

Slot type	Address Parameter Description	Address Acronym (A)	Slot Range (S)
	PV HI limit	PH[4]	1–16
	PV LO limit	PL[4]	1–16
	Deviation HI limit	DH[4]	1–16
	Deviation LO limit	DL[4]	1–16
	Computer set point	CS	1–16
	Computer output	CO	1–16
	Z-input	Z (with a space, read only)	1–16
	Bias (algo 02, 03, and 04)	BI[5]	1–16
	Ratio (algo 02, 03, and 04)	RA[5]	1–16
	Manual PV (algo 20 and 30)	MV[5]	1–16
	Select input (algo21, 26, and 31)	IP[5]	1–16
	Segment number (algo 34)	SN[5]	1–16

Related topics

"Point displays for Honeywell TDC 3000 Data Hiway" on page 68

Addressing a Multifunction Controller (MFC and AMC) for a Honeywell TDC 3000 Data Hiway

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`

Part	Description
`A`	Is the address acronym listed in "Table 3: MC Modulating slot addresses".
`S`	Is the slot number. See "Table 3: MC Modulating slot addresses" 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)

4 Only PH/PL or DH/DL is valid for any given subslot, depending on the slot configuration.

5 These parameters should only be used with the specified algorithms.

Table 3: MC Modulating slot addresses

Address Parameter Description	Address Acronym (A)	Slot Range (S)
Analog point		
Process variable	PV (read only)	1–16
Set point	SP	1–16
Output	OP	1–16
Mode	MD	1–16
Gain	K (with a space)	1–16
Integral time	T1	1–16
Derivative time	T2	1–16
Digital filter	TD	1–16
PV HI limit	PH[6]	1–16
PV LO limit	PL[6]	1–16
Deviation HI limit	DH[6]	1–16
Deviation LO limit	DL[6]	1–16
Computer set point	CS	1–16
Computer output	CO	1–16
Bias	BI[7]	1–16
Ratio	RA[5]	1–16

MC Point Card File Assembly (PCFA) slot addresses

The address format is:

ASs

Part	Description
A	Is the address acronym, listed in “Table 4: MC PCFA slot addresses”.
S	Is the slot number. See “Table 4: MC PCFA slot addresses” for the valid range.
s	Is the subslot number. See “Table 4: MC PCFA slot addresses” for the valid range.

Attention

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)

[6] Only PH/PL or DH/DL is valid for any given subslot, depending on the slot configuration.

[7] The Ratio and Bias parameters should only be used with the PID Ratio and Bias algorithms.

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	17–32	1–16
DO	Digital output	DO	17–32	1–8
	Mode	MD	17–32	1–8
Analog point				
AI	Process variable	PV	23–24	1–8
	Set point	SP	23–24	1–8
	PV HI limit	PH[8]	23–24	1–8
	PV LO limit	PL[8]	23–24	1–8
	Deviation HI limit	DH[8]	23–24	1–8
	Deviation LO limit	DL[8]	23–24	1–8
CI	Process variable	PV	17–32	1–4
	Set point	SP	17–32	1–4
AO	Output	OP	17–32	1–4
	Mode	MD	17–32	1–4
Accumulator point				
CI	Process variable	PV	17–32	1–4

Sequence slot addresses

The points that address the sequence slots are also called CL/MC Process Module Points. See the topic titled "Process module points" for more information. The address format is:

`SEQ`*`nn variable`*

Part	Description
nn	Is the two-digit sequence slot number.
variable	Is the sequence variable listed in “Table 5: MC Sequence slot addresses”.

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

[8] Only PH/PL or DH/DL is valid for any given subslot, depending on the slot configuration.

Sequence Variable	Description
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:

An

Part	Description
A	Is the address acronym listed in “Table 6: MC box variable addresses”.
n	Is the Parameter number. See “Table 6: MC box variable addresses” for the valid range.

Examples

- *LG116* (Logic block 116)
- *NV27* (Numeric 27)

Attention
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	1–128
Flag variable	FG	1–256
Analog point		

Address Parameter Description	Address Acronym (A)	Parameter Number (n)
Numeric variable	NV	1–88
Timer variable	TV	1–32
Timer set point	TS	1–32

Related topics

"Process module points" on page 72

Addressing a Process Interface Unit (PIU) for a Honeywell TDC 3000 Data Hiway

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 consist of up to 32 slots of analog inputs.

Attention

Subslot 8 for slots 16 and 32 are not available as AI.

The address format is:

ASs

Part	Description
A	Is the address acronym listed in "Table 7: HLPIU addresses" and "Table 8: LLPIU and LEPIU address parameters and slot range".
S	Is the slot number. See "Table 7: HLPIU addresses" and "Table 8: LLPIU and LEPIU address parameters and slot range".
s	Is the subslot number. See "Table 7: HLPIU addresses" and "Table 8: LLPIU and LEPIU address parameters and slot range".

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	1–32	1–16
Notify Status	Digital input	DI	1–32	1–16
Sequence of Events	Digital input	DI	1–32	1–16
Pulsed Output	Digital output	DO	1–32	1–8
Latched Output	Digital output	DO	1–32	1–8
	Mode	MD	1–32	1–8
Analog point				

Slot type	Address Parameter Description	Address Acronym	Slot Range	Subslot Range
AI	Process variable	PV	1–32	1–8
	Set point	SP	1–32	1–8
	PV HI Limit	PH[9]	1–32	1–8
	PV LO Limit	PL[9]	1–32	1–8
	Deviation HI Limit	DH[9]	1–32	1–8
	Deviation LO Limit	DL[9]	1–32	1–8
CI (16-bit)	Process variable	PV	1–32	1–8
	Set point	SP	1–32	1–8
CI (32-bit)	Process variable	PV	1–32	1–4
	Set point	SP	1–32	1–4
AO	Output	OP	1 to 32	1–4
	Mode	MD	1 to 32	1–4
Accumulator Point				
AI	Process variable	PV	1 to 32	1–8
CI (16-bit)	Process variable	PV	1 to 32	1–8
	Set point	SP	1 to 32	1–8
CI (32-bit)	Process variable	PV	1–32	1–4
	Set point	SP	1–32	1–4

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	1–32	1–8
	PV HI limit	PH[10]	1–32	1–8
	PV LO limit	PL[10]	1–32	1–8
	Deviation HI limit	DH[10]	1–32	1–8
	Deviation LO limit	DL[10]	1–32	1–8
	Set point	SP	1–32	1–8
Accumulator point				
AI	Process variable	PV	1–32	1–8

Addressing a Data Hiway Port (DHP) and an LCS620 HIM for a Honeywell TDC 3000 Data Hiway

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.

An LCS620 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 LCS620 is the same as for the DHP.

9 Only PH/PL or DH/DL is valid for any given subslot, depending on the slot configuration.

10 Only PH/PL or DH/DL is valid for any given subslot, depending on the slot configuration.

The address format is:

`ASs`

Part	Description
A	Is the address acronym listed in “Table 9: DHP/HIM addresses”.
S	Is the slot number. See “Table 9: DHP/HIM addresses” for the valid range.
s	Is the subslot number. See “Table 9: DHP/HIM addresses” 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	1–15, 17–31	1–16
Latched Output	Digital output	DO	1–15, 17–31	1–8
	Mode	MD	1–15, 17–31	1–8
Analog Point				
AI	Process variable	PV	1–15, 17–31	1–6
	Set point	SP	1–15, 17–31	1–6
	PV HI limit	PH[11]	1–15, 17–31	1–6
	PV LO limit	PL[4]	1–15, 17–31	1–6
	Deviation HI limit	DH[4]	1–15, 17–31	1–6
	Deviation LO limit	DL[4]	1–15, 17–31	1–6
CI	Process variable	PV	1–15, 17–31	1–8
AO	Output	OP	1–15, 17–31	1–4
	Mode	MD	1–15, 17–31	1–4
Accumulator Point				
AI	Process variable	PV	1–15, 17–31	1–6
CI	Process variable	PV	1–15, 17–31	1–8

Addressing generic hardware for a Honeywell TDC 3000 Data Hiway

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 the topic titled "Other documentation for Honeywell TDC 3000 Data Hiway" 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.

[11] Only PH/PL or DH/DL is valid for any given subslot, depending on the slot configuration.

Attention

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'xxyy B WW format

Part	Description
xx	Slot number minus 1 in octal (for example, slot *10 = O'11*)
yy	Address in octal as defined in the TDC manuals
B	Starting bit number
WW	Width from the starting bit
format	Data format (as defined in the TDC manuals): • B0 • B4 • B7 • B12 • B15 • B12E • B12S

Example

H:O'0716 0 16 B4 (For slot 8, MFC modulating slot auxiliary constant at NN16 with format B4).

Related topics

"Other documentation for Honeywell TDC 3000 Data Hiway" on page 9

Linearization algorithms for Honeywell TDC 3000 Data Hiway

When a non-linear input is selected as part of the TDC controller configuration, the server automatically applies a PV algorithm to the point to linearize the PV and alarm limit values using the appropriate piece-wise, fourth-order polynomial algorithm. These algorithms are similar to the TDC Engineering Unit Conversion (EUC) functions performed by the TDC operator stations.

Attention

For linearization to take place, the server point must not have any other PV algorithm attached.

The PV algorithm numbers reserved for linearization are 101 to 115 and 121 to 135 corresponding to the TDC EUCs 1 to 15 in degrees Fahrenheit and degrees Celsius respectively.

Table 10: Linearization algorithms

PV algo	TDC EUC	Description	Valid Range (°F)	Valid Range (°C)
101/121	1	Type J thermocouple	-103 to 2003	-75 to 1095
102/122	2	Type K thermocouple	-103 to 2505	-75 to 1374
103/123	3	Type T thermocouple	-303 to 752	-186 to 400
104/124	4	Type S thermocouple	-2 to 3203	-19 to 1762
105/125	5	Square root	N/A	N/A

PV algo	TDC EUC	Description	Valid Range (°F)	Valid Range (°C)
106/126	6	Platinum RTD Burns	-303 to 1202	-186 to 650
107/127	7	RH radiamatic	1097 to 3203	592 to 1762
108/128	10	Type B thermocouple	1468 to 3099	798 to 1704
109/129	11	Type E thermocouple	-331 to 1835	-202 to 1002
110/130	12	Type R thermocouple	-58 to 3200	-50 to 1760
111/131	13	Type R thermocouple (JIS)	32 to 3218	0 to 1770
112/132	14	Platinum RTD DIN	-349 to 1200	-212 to 649
113/133	15	Platinum RTD JIS	-382 to 1166	-200 to 630

Point displays for Honeywell TDC 3000 Data Hiway

The following TDC displays are installed as part of the Station software:

Display Filename	Description
sysDtlAna_TDC_ECPID.htm	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.htm	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 the section below titled "Point configuration for TDC EC PIDRB display."
sysDtlAna_TDC_EC.htm	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 the section below titled "Point configuration for TDC EC display."

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*	
Auxiliary	A1 Parameter Name	Ratio	
	A1 Source & Destination Address	See the topic titled "Addressing an Extended Controller (EC) for a Honeywell TDC 3000 Data Hiway."	
	A2 Parameter Name	Bias	
	A2 Source & Destination Address	See the topic titled "Addressing an Extended Controller (EC) for a Honeywell TDC 3000 Data Hiway."	

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.

Attention

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.

Table 11: Primary PV/X Input Point

Tab	Property	Entry	Notes
Main	PV Source Address	See the topic titled "Addressing an Extended Controller (EC) for a Honeywell TDC 3000 Data Hiway."	PV/X input source address
	Engineering Units	As required	PV/X input engineering units
Display	Point Detail Display	`sysDtlAna_TDC_EC`	
Control	SP Source & Destination Address	Leave blank	
	OP Source & Destination Address	See the topic titled "Addressing an Extended Controller (EC) for a Honeywell TDC 3000 Data Hiway."	
	MD Source & Destination Address	See the topic titled "Addressing an Extended Controller (EC) for a Honeywell TDC 3000 Data Hiway."	
Auxiliary	A1 Parameter Name	YinXeu	
	A1 Source & Destination Address	SP parameter address. See the topic titled "Addressing an Extended Controller (EC) for a Honeywell TDC 3000 Data Hiway."	This value is used by the Y input bar
User Defined	Parameter Name	Yinput	
	Link Type	Parameter Reference	
	Point Name	Enter secondary point name	Primary point name with suffix Y
	Parameter Name	SP	

Table 12: Secondary SP/Y Input Point

Tab	Property	Entry	Notes
Main	Point ID	Primary point ID with suffix Y	PV/X input source address
	Engineering Units	As required	SP/Y input engineering units
Display	Point Detail Display	default	
Control	SP Source & Destination Address	See the topic titled "Addressing an Extended Controller (EC) for a Honeywell TDC 3000 Data Hiway."	

Tab	Property	Entry	Notes
	OP Source & Destination Address	Leave blank	
	MD Source & Destination Address	Leave blank	

Related topics

“TDC point property settings” on page 55
“Addressing an Extended Controller (EC) for a Honeywell TDC 3000 Data Hiway” on page 59

Control Language reference

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 SCADA Controllers 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.

Related topics

"Configuration tasks" on page 72
"Program development" on page 75
"Control Language basics" on page 82
"Statements" on page 102
"Structures" on page 120
"Syntax summary" on page 138
"Additional tasks for Honeywell TDC 3000 Data Hiway" on page 42

Configuration tasks

This section describes the configuration tasks you can perform on CL programs.

Related topics

"CL acronym libraries" on page 72
"Process module points" on page 72
"C Link addresses" on page 73
"Logic blocks" on page 73

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 four libraries. Each MC is assigned to one CL Acronym Library as part of the Controller configuration. See the topic titled "Main properties for a Honeywell TDC 3000 Data Hiway controller" for more information. The default allocation for all MCs is library 1. The CL Acronym Library assignment is displayed on the MC SCADA Controllers 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	129–256, 513–576
Phase Names	129–256, 513–576
Step Labels	257–512
Send String item	1–576

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.

Attention

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.

Related topics

"Main properties for a Honeywell TDC 3000 Data Hiway controller" on page 38

Process module points

The CL programs are monitored and controlled via points called *process module* (PM) points. Multiple CL programs can be compiled against a single *process module* 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 as 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

Related topics

"Addressing a Multifunction Controller (MFC and AMC) for a Honeywell TDC 3000 Data Hiway" on page 60

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.

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 SCADA Controllers 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 recompiled.

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 about configuring *logic blocks*, see the *SOPL User's Manual*.

Program development

This section describes the CL software environment and program directories, and contains information about creating and troubleshooting CL programs.

Related topics

"CL software environment" on page 75
"CL program directories" on page 76
"Creating CL programs" on page 76
"CL sequence messages" on page 77
"Compiling CL programs" on page 77
"Compile time errors" on page 78
"Downloading sequence programs" on page 78
"Troubleshooting sequence programs" on page 79
"Sequence program run time errors" on page 79
"Uploading sequence programs" on page 79
"Comparison with the TPS CL MC compiler" on page 79

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
Maximum number of 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 one item each. Example: (x + y) <6 contains five items.	100 items
Maximum number of declarations in a sequence (includes locals, externals, and constants). All declarations count as one item each.	approximately 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).	10 words to 400 words (average is approximately 20 words)

CL program directories

Default Path	Description
<install folder>\Honeywell\Experion PKS\server\cl[12]	The CL Compiler must be executed from this directory.
<install folder>\Honeywell\Experion PKS\server\cl\src[12]	CL source files *.cl*.
<install folder>\Honeywell\Experion PKS\server\cl\lst[12]	CL list files *.ls* and *.xf* generated by the compiler.
<install folder>\Honeywell\Experion PKS\server\cl\pls[12]	Reference files *.d* generated by the compiler.
<install folder>\Honeywell\Experion PKS\server\data\cl[12]	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 table "MC parameters accessible by TDC CL" in the topic titled "Accessing parameters," or see the appendices in the *SOPL User's Manual*.

Related topics

“Accessing parameters” on page 126

[12] Where *<install folder>* is the location where Experion is installed.

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:

To zero a memory address in the controller

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:

   a. `write bb,1476,0000 <`
   b. Check the MC address using the read command:

      `read bbc,1476 <`
   c. To quit the **tdctst** utility type:

      `q <`

   Check that messages are received and can be acknowledged.

Related topics

“Comparing Honeywell TDC 3000 Data Hiway with Experion” on page 13
“Sequence programs for Honeywell TDC 3000 Data Hiway” on page 46

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
`filename`	The source file without the `.cl` extension.
`-d`	Optional switch to compile source lines beginning with %DEBUG. See the topic titled "Embedded compiler directives" for more information.
`-x`	Optional switch to cause the compiler to output cross reference information to a file in the `lst` folder (`filename.xf`).

This results in a listing file `filename.ls` in the `lst` folder.

Attention
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.

Related topics

"Embedded compiler directives" on page 119
"Troubleshooting Honeywell TDC 3000 Data Hiway issues" on page 151

Compile time errors

There are three categories of compile time errors:

- *Fatal Errors* — these errors prevent the compiler from further compilation.
- *Errors* — these include syntax errors and semantic check errors. Detailed messages are included in the `.lst` list file.
- *Warnings* — these errors do not prevent successful compilation. They 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
`hh`	Hiway number (channel number) for the MFC/AMC.
`bb`	Box ID for the MFC/AMC.
`s`	Sequence slot number of the Process module point for the Sequence.
`ppp`	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 SCADA Controllers 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 three digits of the *.mo* file name.
5 Enter the starting Memory Block Number and then press 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 SCADA Controllers 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 Esc key to refresh the display as quickly 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.

Attention
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 that 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\cl* folder. If a file with the same name already exists in this folder, it will be overwritten. The facility can be used to back up 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 from the LCN CL compiler. This section describes the main differences and is included as a comparison for those familiar with the LCN implementation.

Compiler usage

File handling	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 *.cl* 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.

<table>
<tr><td>Identifier</td><td>All identifiers in this TDC CL can be 10 characters long, instead of eight characters for LCN version. Also, the current implementation does not reject identifiers with consecutive break characters. However, an identifier cannot start with a break character.</td></tr>
<tr><td>Reserved words</td><td>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:
<pre>
ABS
LOGICAL
NUMBER
OFF
ON
</pre></td></tr>
<tr><td>Constants & constant expressions</td><td>All constant expressions (that is, expressions involving numerical constants or predeclared 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.</td></tr>
<tr><td>TDC CL statements</td><td>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 anymore. 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.
<pre>
ALLOWED   NOT ALLOWED
LAB1: LOOP LAB1: LOOP
    --   --
    --   --
LAB2: LOOP LAB2: LOOP
    --   --
    --   --
REPEAT LAB2 REPEAT LAB1
    --   --
    --   --
REPEAT LAB1 REPEAT LAB2
    --   --
    --   --
</pre>
Send statement

The Advanced control point (ACP) Send statement is not supported.
</td></tr>
<tr><td>MC variable parameters</td><td>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.</td></tr>
</table>

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 *.ls* 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.

The description of each TDC CL element, statement, and structure is presented in a four-part format, as follows:

Table 13: 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 the section "Syntax" in the topic titled "Rules and elements," the way in which the syntax for an element, statement, or structure is presented is discussed. See the topic titled "Syntax summary" 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.

Related topics

"Rules and elements" on page 82
"Data types" on page 90
"Variables and declarations" on page 93
"Expressions and conditions" on page 96
"Syntax summary" on page 138

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 section below titled "TDC CL restrictions."

Characters can be combined to generate the following basic elements:

- Comments
- Identifiers (including reserved words; see the section below titled "Identifiers")
- 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 and the statements are combined to build structures.

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 the topic titled "Statements") 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 the topic titled "Structures" 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 the topic titled "Syntax summary."

TDC CL restrictions

This section lists the restrictions placed on the language.

Length of identifiers

The following restrictions apply to identifiers (see the section below titled "Identifiers"):

- 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 the table below titled "Variable Characters in ISO 646"). 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 alphabetics in TDC CL. (They can be used in Strings and comments.)

Table 14: 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
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.
```

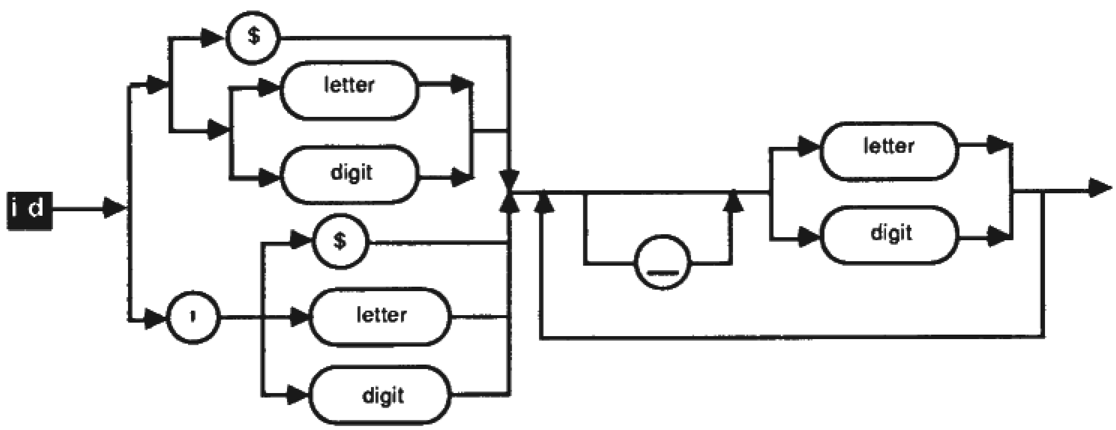
Incorrect examples of comments

```
--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 the "Special identifiers" section below) 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.

Uppercase and lowercase characters can be used in identifiers, but the distinction between cases is not significant; that is, each lowercase character is considered the same as its uppercase 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 the section above titled "Length of identifiers").

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

The table “Reserved Words” shows a list of the TDC CL reserved words (identifiers) that cannot be redefined by any structure.

Table 15: 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
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.

Day_Time	Built-in function
Exp	Built-in function
Int	Built-in function
Ln	Built-in function
Log10	Built-in function
Max	Built-in function
Min	Built-in function
Round	Built-in function
Set_Time	Built-in subroutine
Sqrt	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.

```
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.

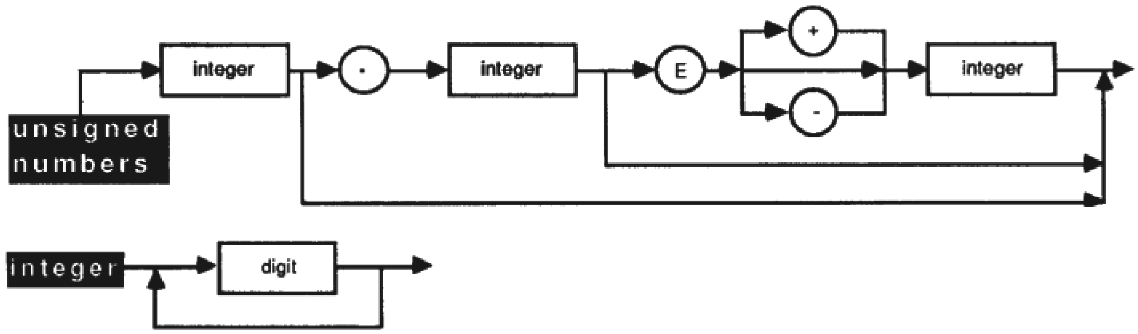
Scopes can sometimes be nested. A Step must be within a Phase, and a Phase within a Sequence Program. Local Subroutines of Sequence Program are considered to be nested within that program unit.

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.

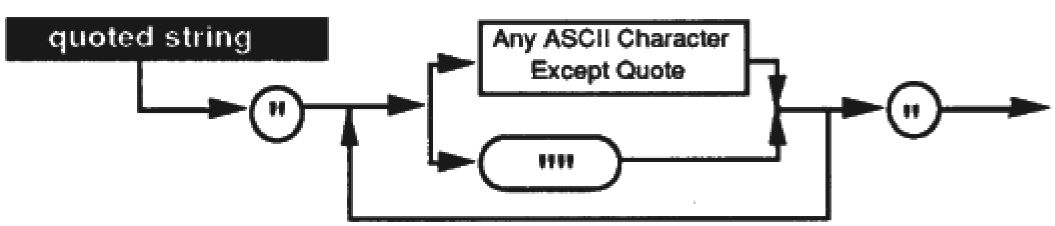
Numbers examples

```
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 the “SEND statement” section in the “Program statement definitions” topic).

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 the table “Special symbols” are special symbols.

Table 16: Special symbols

Symbol	Meaning
+ - * / **	Arithmetic operators
< <= > >= <>	Relational operators
=	Equality, assignment operator
..	Range separator
( )	Parentheses
: ; . ,	Punctuation
'	String separator
-	Comment separator
&	Line continuation
'	Special identifier prefix

Related topics

“Statements” on page 102
“Structures” on page 120
“Program statement definitions” on page 102

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 the topic titled “Structures” for more information.

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 the table below titled "Pascal Boolean vs. TDC CL Logic" should show this difference.

Table 17: 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

```
coeff (6) -- one dimension
pump(x+y*z) -- indexed by expression
```

String data type

String variables are not supported; string literals (quoted strings) are allowed in only the SEND statement.

Related topics

“Structures” on page 120

Variables and declarations

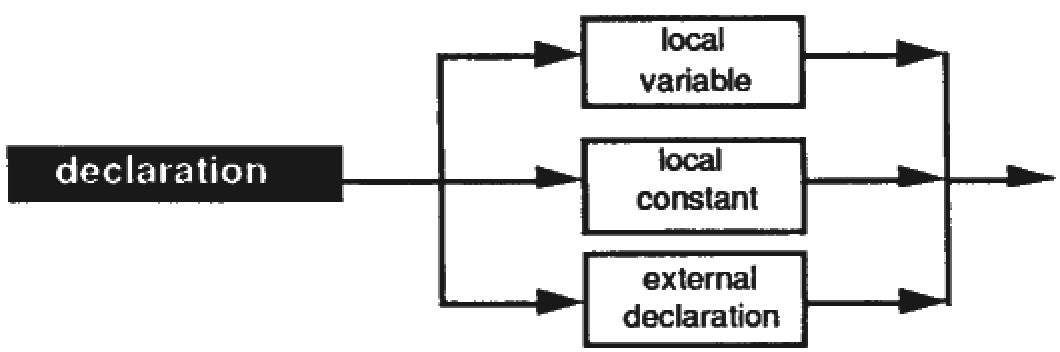
Overview

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

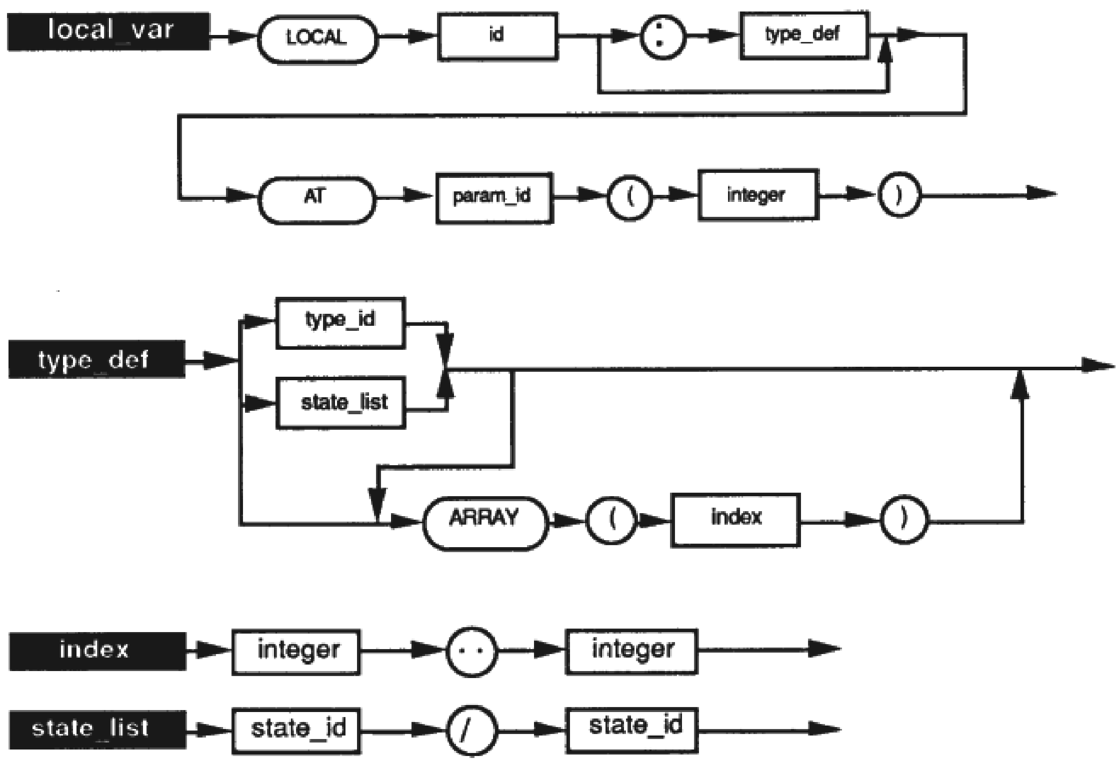


Local variables

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.

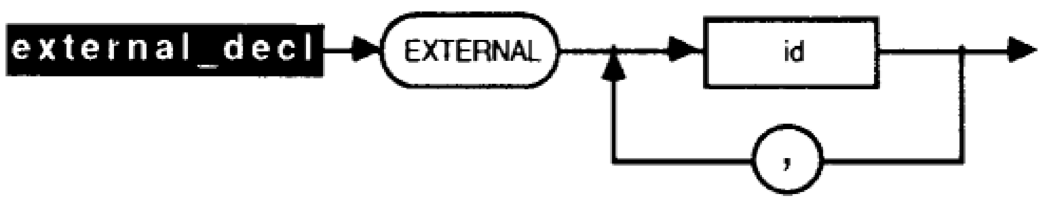
Example

```
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 the section above titled "External data points."

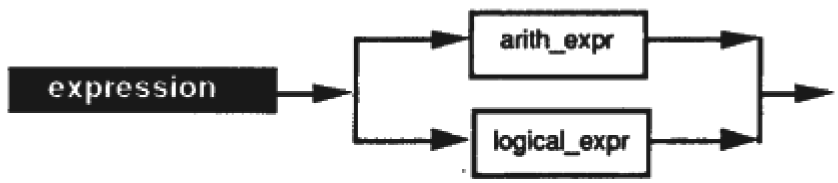
Certain parameters of the related box Data point are made to appear (to the TDC CL compiler) as parameters of the process module Data point. This enables convenient access to box numerics, timers, flags, I/O slots, and controller slots.

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.

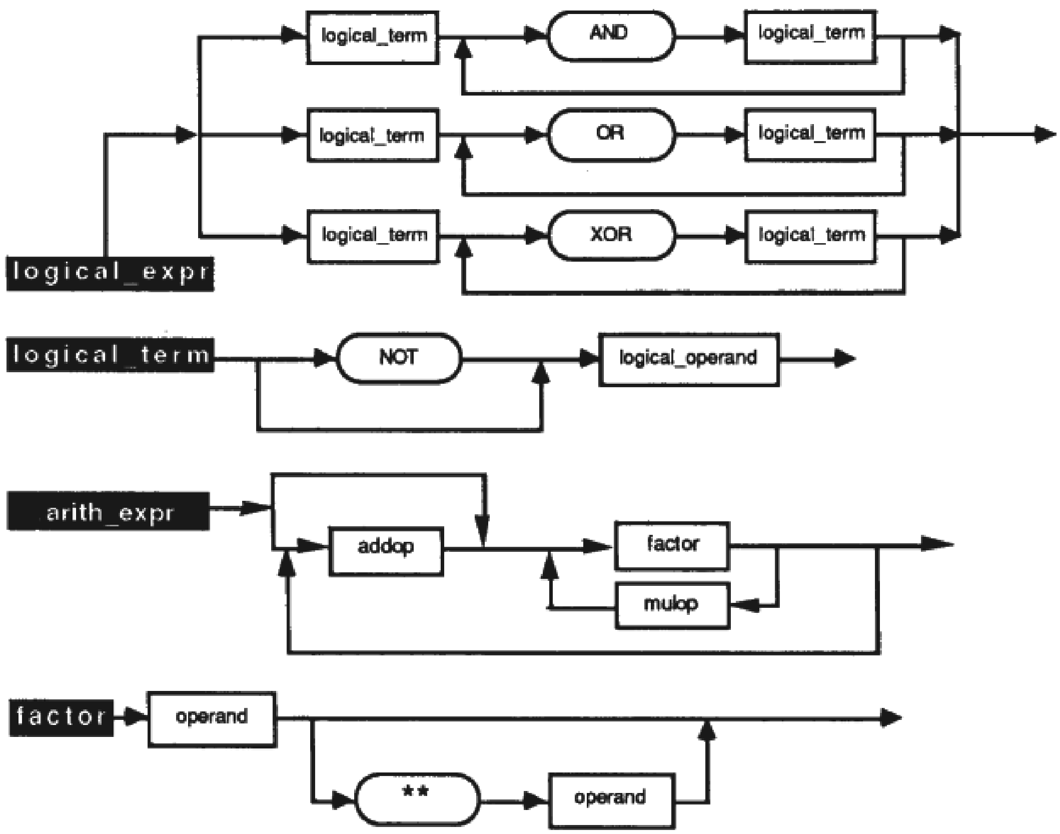
Expressions and conditions syntax



Arithmetic and logical expressions

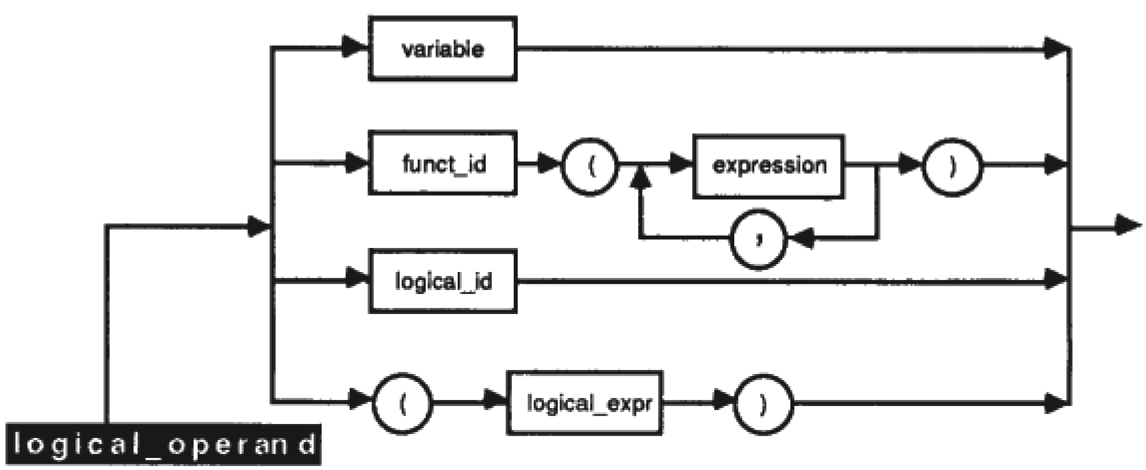
An expression is a formula that defines the computation of a value. The components of an expression are operands and operators.

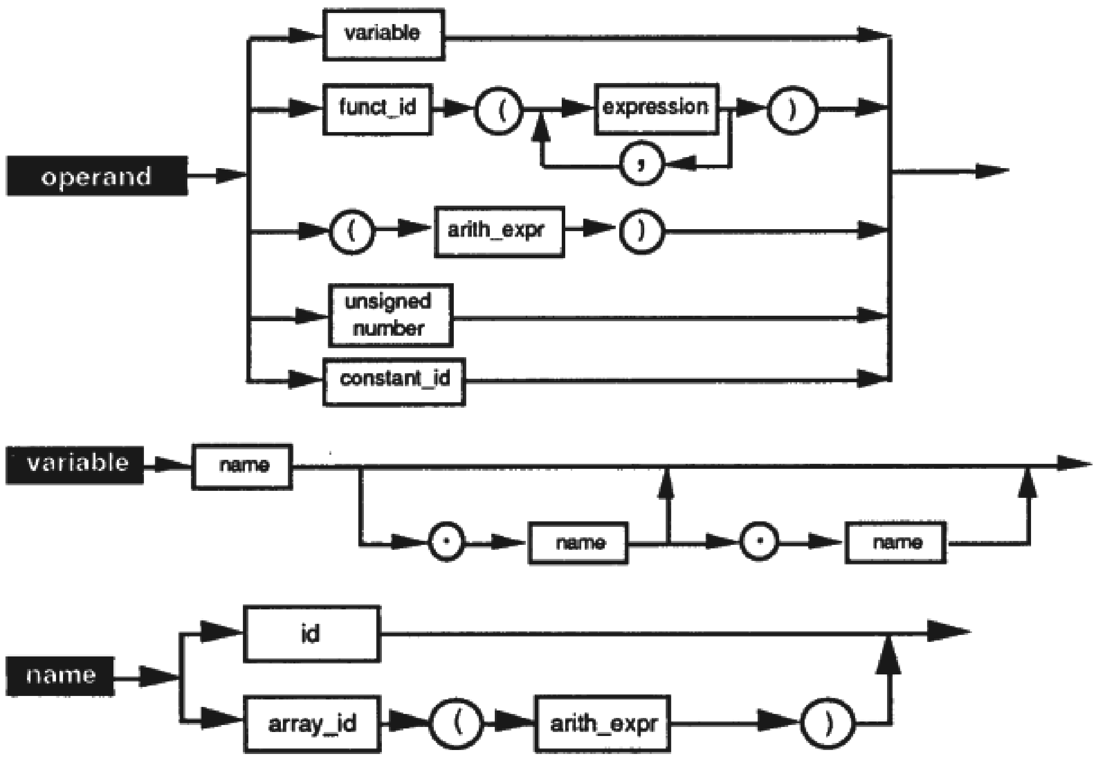
Syntax



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 the table below titled "Operator Priorities"; higher priority means closer binding.

Table 18: Operator Priorities

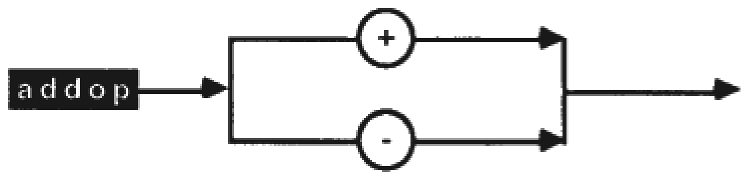
Priority	Operator	Meaning
3	**	Exponentiation
2	*	Multiplication
	/	Division
	MOD	Remainder
	NOT	Logical complement
1	+	Sum
	-	Difference, negation
	AND	Logical AND
	OR	Logical OR
	XOR	Logical exclusive OR

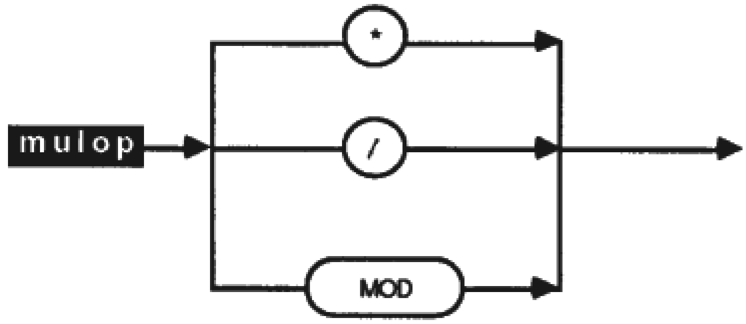
For example, in the expression a + b * c, b * c is evaluated first because the * operator has a higher priority. Operators with the same priority are evaluated left to right.

Arithmetic operators definition

The arithmetic operators are +, -, *, /, MOD and **; their usual meanings are as listed in the table "Logical Operators Truth Table." 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: $X ** Y = X^{Y} = e^{y**1nx}$, $X > 0$
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 the table below titled "Logical Operators Truth Table." Logical operators can only take operands of type LOGICAL.

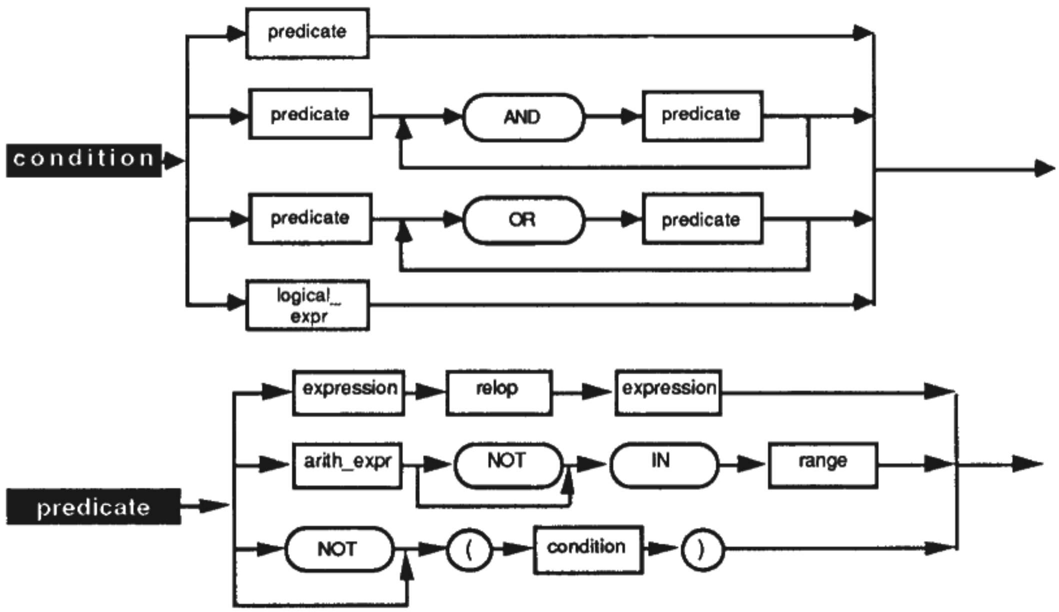
Table 19: Logical Operators Truth Table

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 the table below titled "Relational Operators."

Table 20: Relational Operators

Operator	Meaning
<	Less than
=	Equal to
>	Greater than
£	Less than or equal to
Š	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 AND b = 6 OR c > 7
```

is ambiguous and must be rewritten as one of the following:

```
(a < 5 AND b = 6) OR c > 7
a < 5 AND (b = 6 OR c > 7)
```

Statements

This section describes the statements that you can use when building TDC CL structures.

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*

  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*

  Cause the program to wait for some event to occur or for a time delay: PAUSE and WAIT.
- *Termination statements*

  Signify the termination of the program or of a part of it: FAIL, EXIT, ABORT, and END.
- *Communication statements*

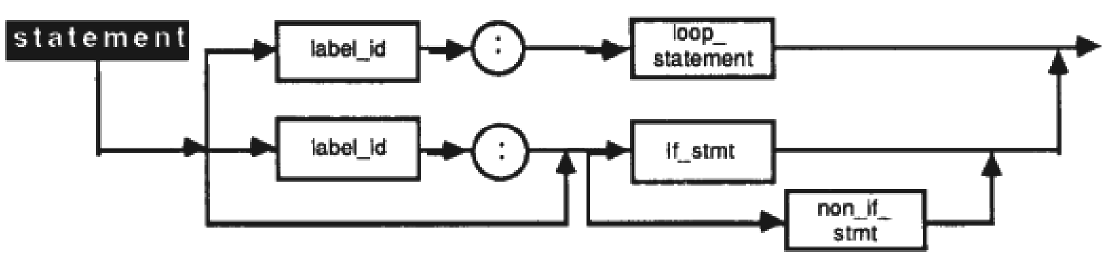
  Communicate with an operator: SEND.

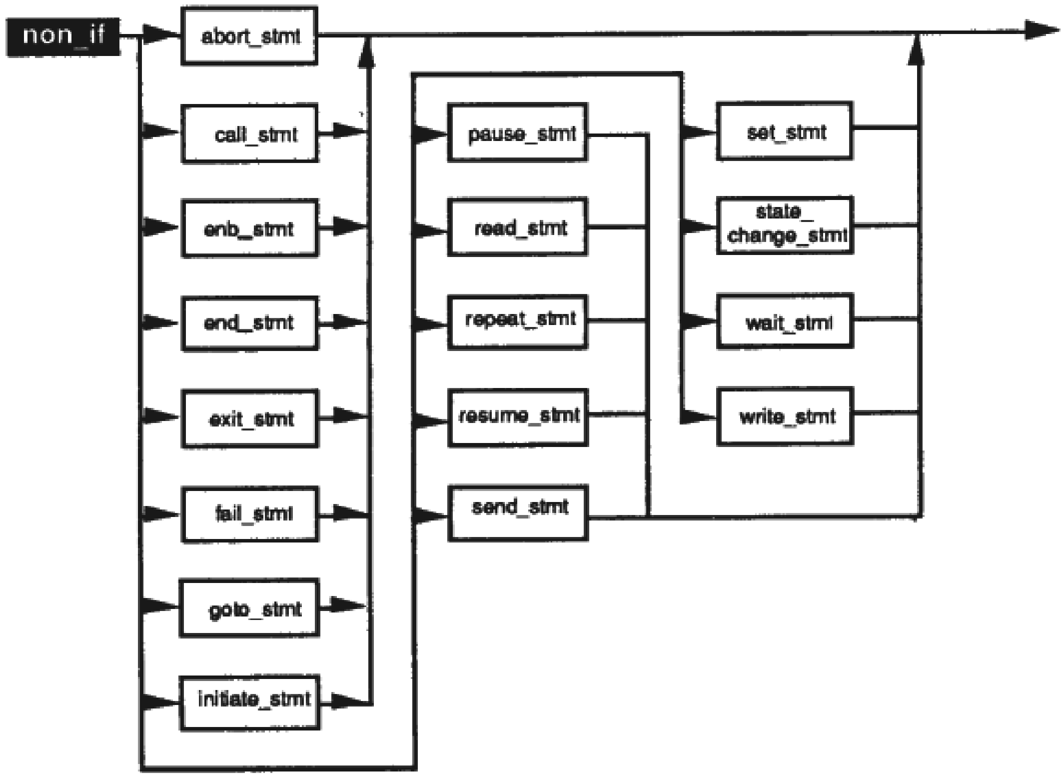
Related topics

"Program statement definitions" on page 102
"Embedded compiler directives" on page 119
"Rules and elements" on page 82
"Structures" on page 120
"Abnormal condition handlers" on page 123

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

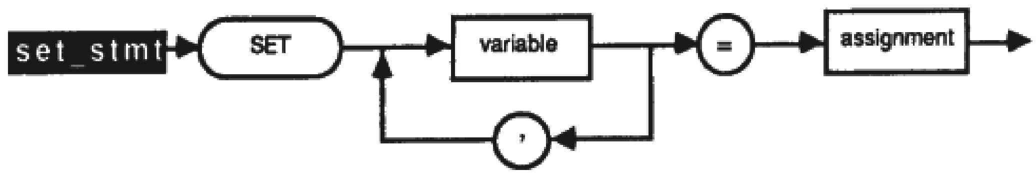
```
lab_01:     CALL test (x, y, z)
IF x >y THEN (SET x= z;
& badlabel :  SET z = y) – NOT VALID
```

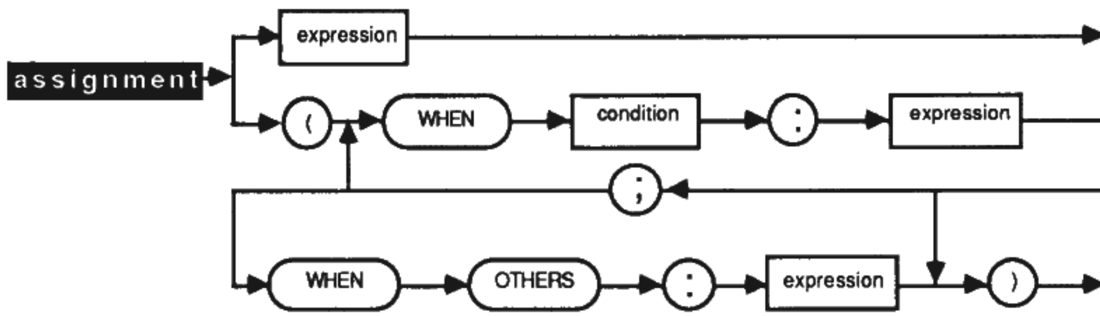
In this example, `badlabel` appears on a continuation line; therefore, it is invalid.

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.

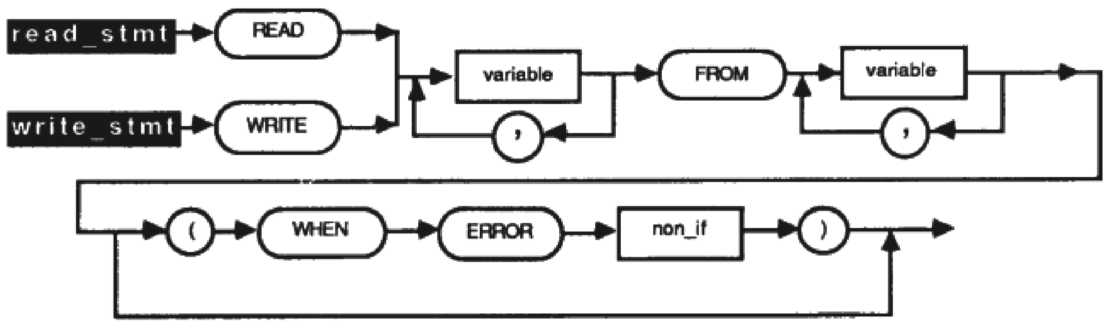
Example

```
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, `$HyxxByy`) must be included in the EXTERNAL declarations. You should read all of the sections in the section "External data point parameters" in the topic titled "Accessing parameters" 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 21: 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	-	-	-	-

Type of Point	OP	PV	PVP	STATE	SP	SPP
Numeric	-	R/W	-	-	-	-
Regulatory control	X	R	R	-	R	R
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 templ : 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 timerl.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.

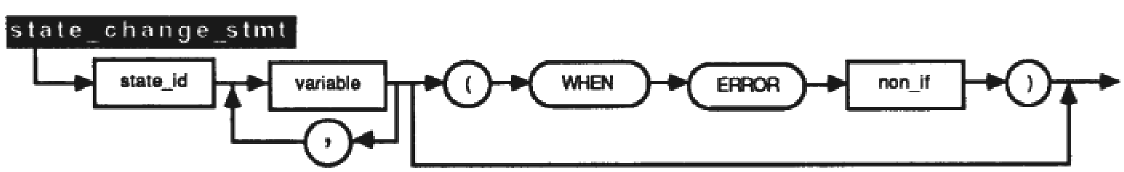
In the MFC, if a communication error occurs during a READ, one or more destination variables remain unchanged.

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          Output states are start/stop
Valve2,Valve3   Output states are open/close
37SW            Output states are low/high
```

Then:

```
stop MOTOR1
open Valve2, Valve3
high 37SW (WHEN ERROR GOTO STEP retry)
```

are valid STATE CHANGE statements.

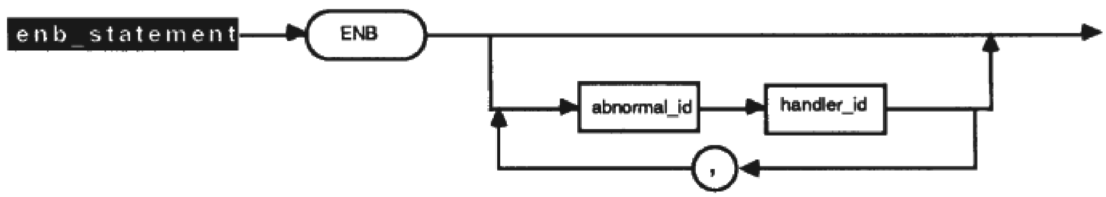
ENB statement

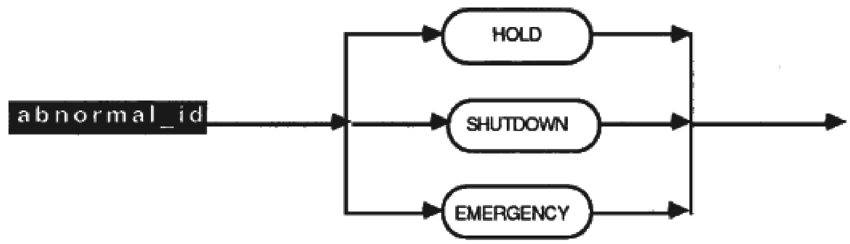
This statement enables a new set of abnormal condition handlers, or disables all currently enabled abnormal condition handlers.

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





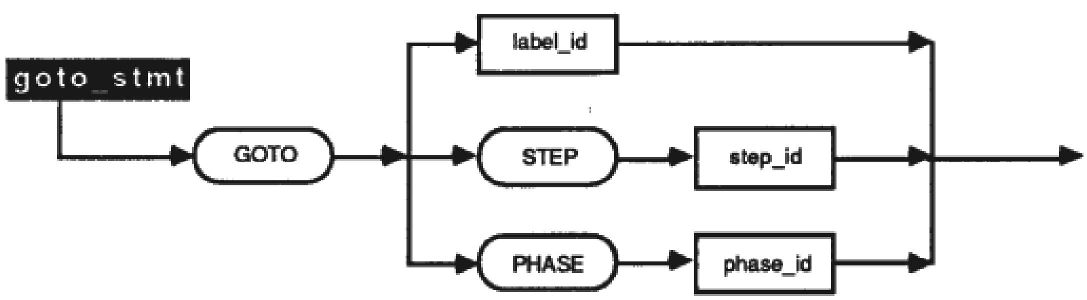
Example

```
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.

Attention

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.

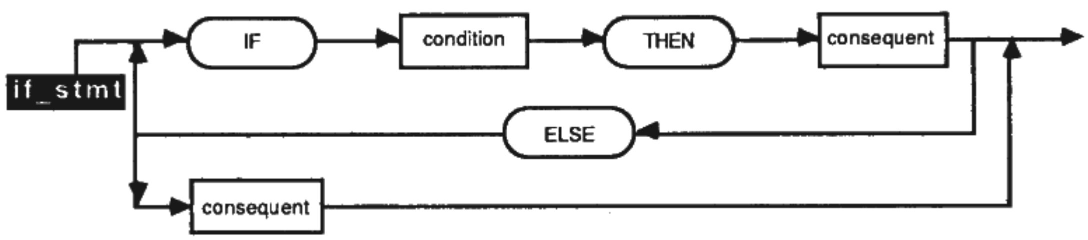
Example

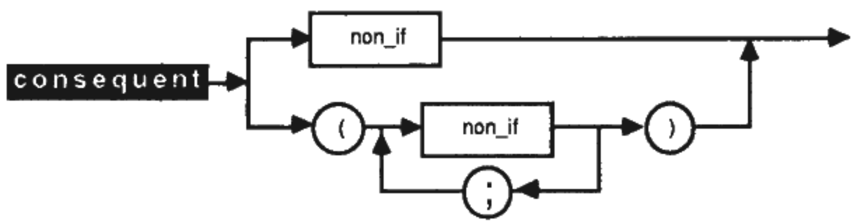
```
GOTO label
    – simple GOTO
    . . .
PHASE test
    – each execution of 'GOTO PHASE test' causes a phase
    – change to be placed in the MFC's Sequence
    . . .
GOTO PHASE test
    – Circular List
PHASE test
    – this brings you back to beginninglabel:
    – of the code for this phase without 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 “Figure 19: IF Statement Flowchart”.

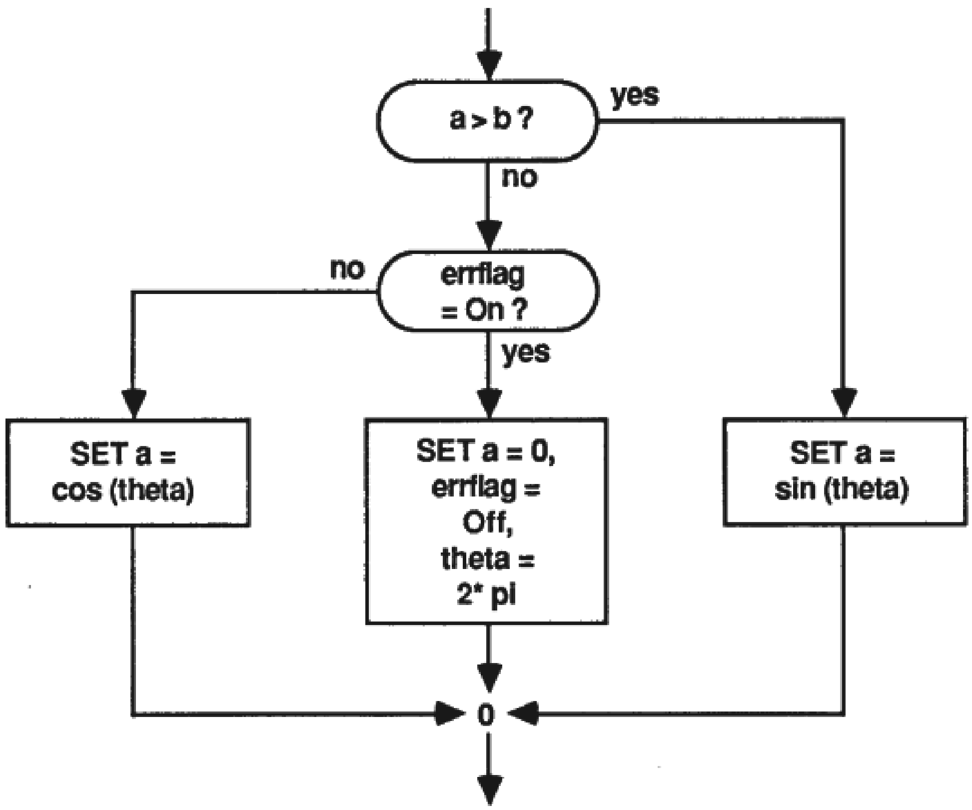
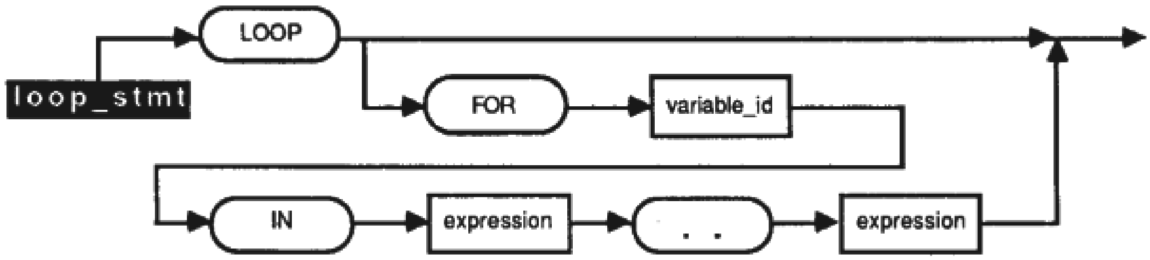


Figure 19: IF Statement Flowchart

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 reevaluated 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

```
label :   LOOP
label :   LOOP FOR count IN 10..20
```

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.

Attention
A REPEAT statement is a preemption 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.

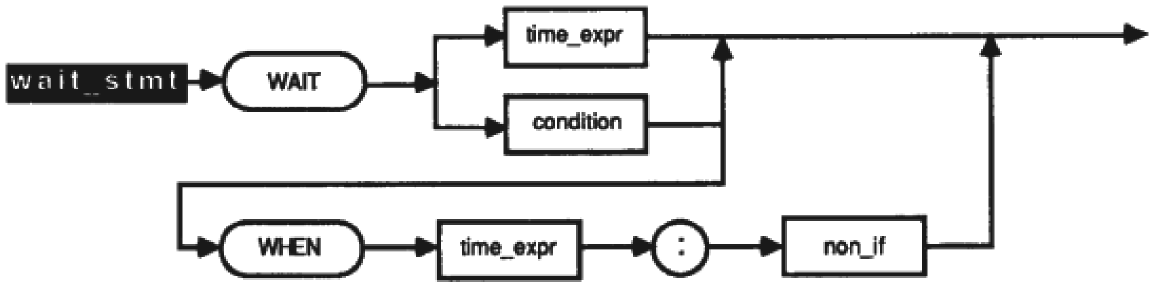
This statement is a Preemption point.

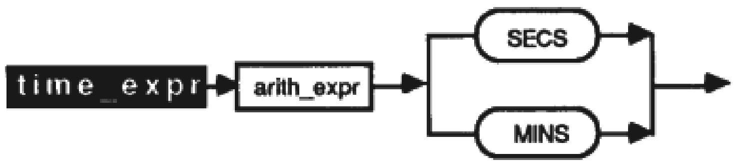
Syntax

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

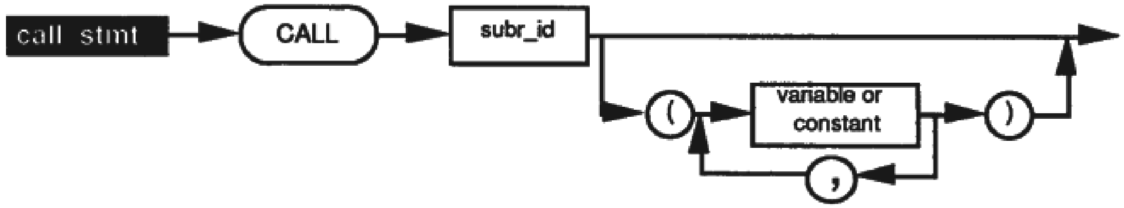
```
WAIT 5 MINS
WAIT x >5.5
WAIT A_001.PV IN A_001.lolim .. A_001.hilim
    boil:LOOP FOR tick IN 1..60 – minutes
        IF temp >= 100 THEN GOTO done
        WAIT press > 1(WHEN 1 MINS: GOTO next)
            ...
            – do something to adjust pressure
    next: REPEAT boil
            – go test 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.

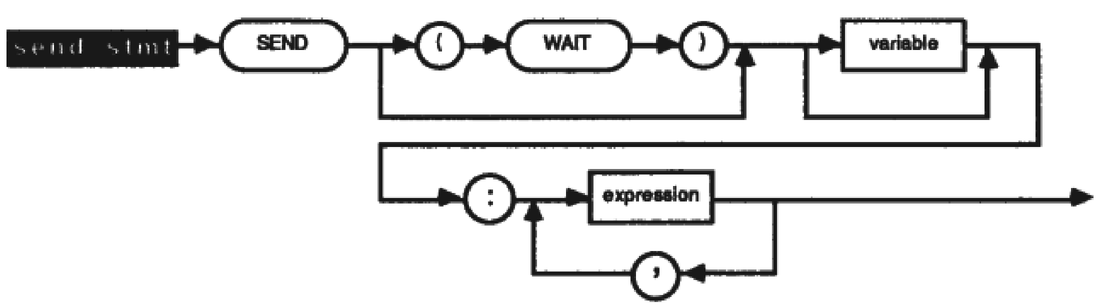
Example

```
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.

Example

```
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'
```

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.

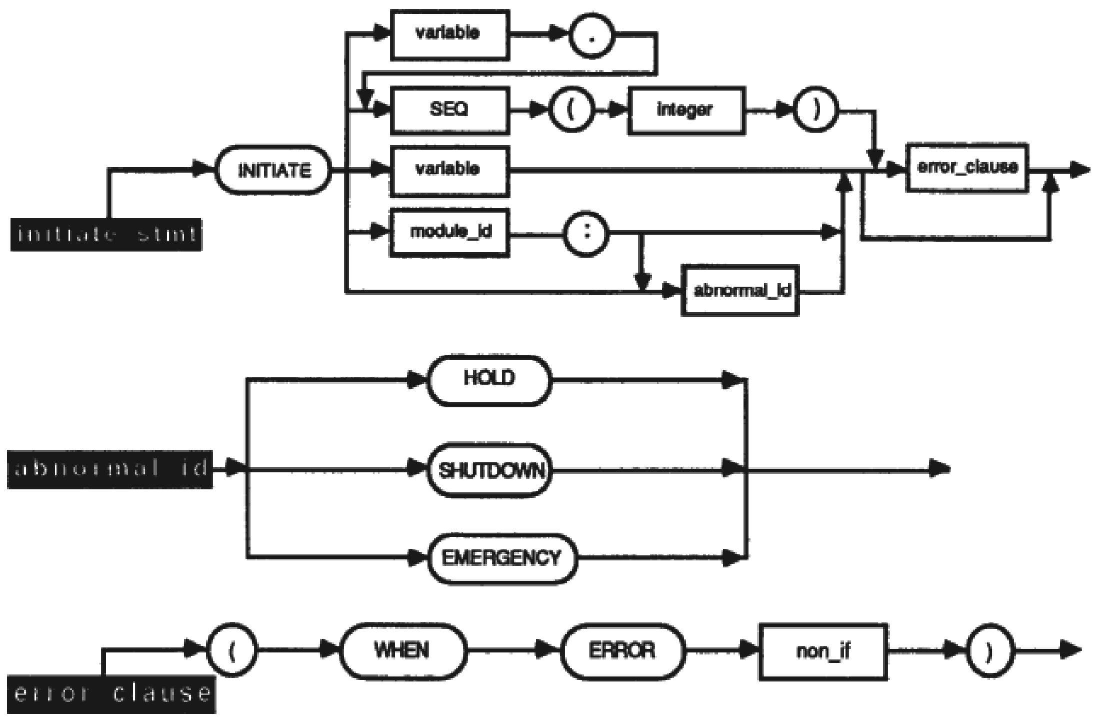
This is done for two reasons. First, the handler may need to execute a SEND with WAIT. Second, it is likely that when the handler is done, the program will no longer need the confirmation that it was waiting for.

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 the section “Box data point identifiers” in the “Rules and elements” topic) 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

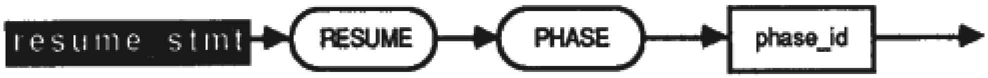
Example

```
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

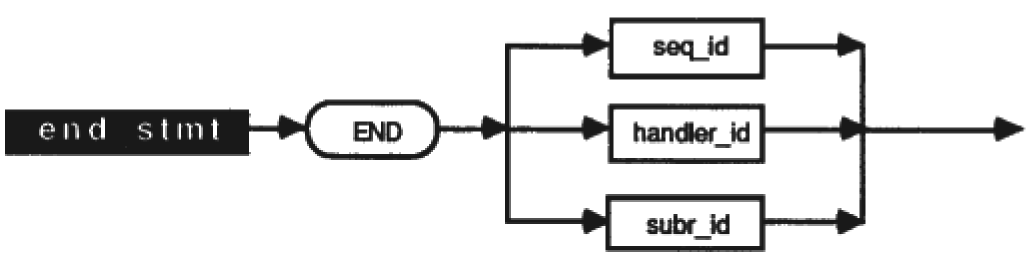
This statement causes program termination. When executed in a subroutine, it terminates both the subroutine and its caller.

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.

Related topics

“Rules and elements” on page 82

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 uppercase or lowercase.

%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.

Related topics

“Compiling CL programs” on page 77

Structures

This sections has information specifically about TDC CL for the Multifunction Controller.

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 the topic titled “Statements.”

Related topics

“Sequence program definition” on page 120
“Abnormal condition handlers” on page 123
“Restart routines” on page 125
“Accessing parameters” on page 126
“User-written subroutines” on page 134
“Built-in functions and subroutines” on page 136
“Data types” on page 90
“Rules and elements” on page 82
“Statements” on page 102

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 the topics titled "User-written subroutines" and "Built-in functions and subroutines."

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.

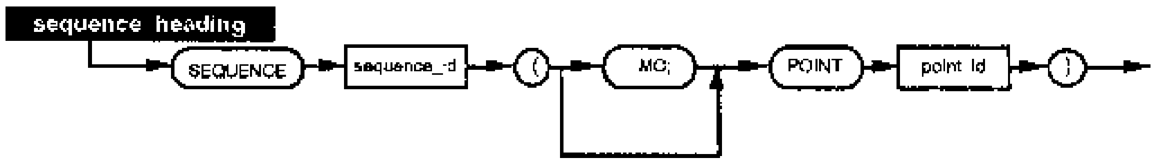
The main sequence is made up of phases; a phase or subroutine consists of one routine. An abnormal condition handler consists of one or two routines, the second (if present) being its restart routine. Routines (phases, handlers, and subroutines) are made up of steps; However, if a routine consists of 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

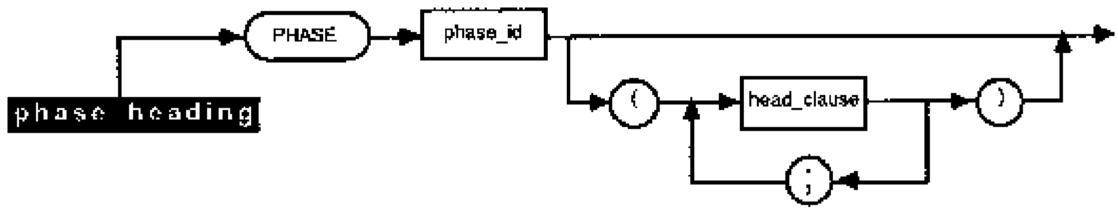
A phase is a major process milestone. Phase boundaries are key synchronization points in the sequence program. The PHASE heading identifies the beginning of a new phase and sets up the operating conditions for the execution of its routines.

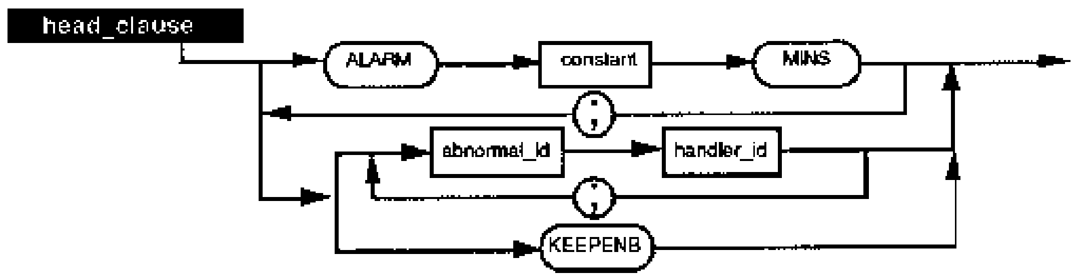
During a phase, a check is made at each Preemption point for the occurrence of abnormal conditions. When such a condition occurs, all activities cease and an Abnormal condition handler is activated.

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

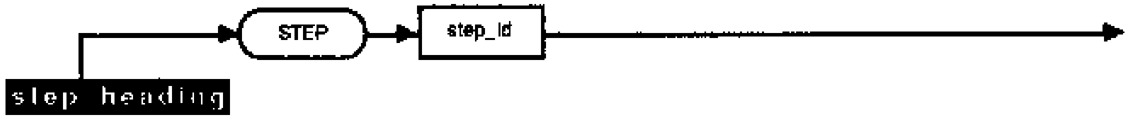
```
PHASE fill_up (ALARM 2 MINS;
&   HOLD fillhold)
PHASE baking (ALARM 30 MINS;
&   SHUTDOWN oven_clr
&   EMERGENCY oven_stp)
```

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 the topic titled "Statements" 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
```

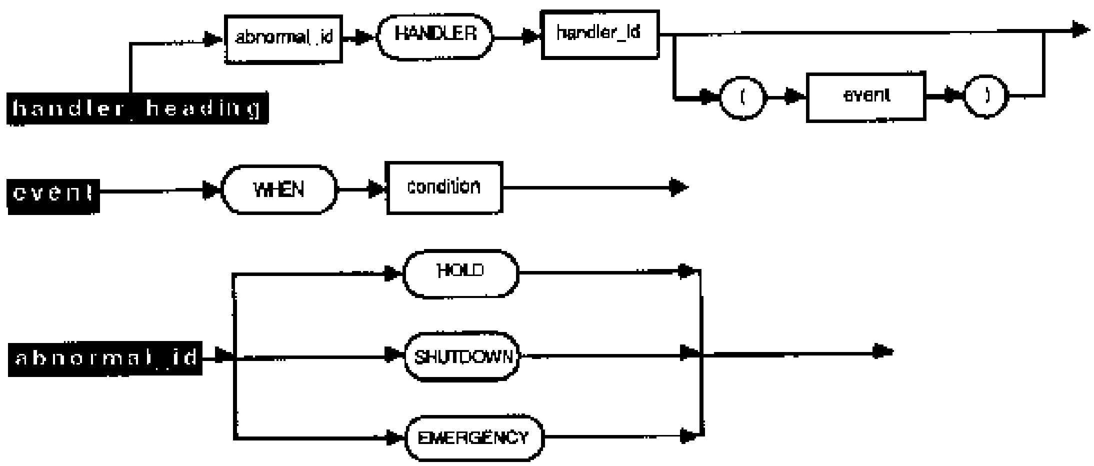
If the operator, or any INITIATE statement (in the same or another program), should attempt to start an Abnormal condition handler that is not enabled, the condition is Propagated to the next lower priority. That is, if an attempt is made to start EMERGENCY and there is no enabled EMERGENCY handler, 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 (KEEPENB)
        ....
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)
```

Related topics

"Statements" on page 102

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 reentry 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.

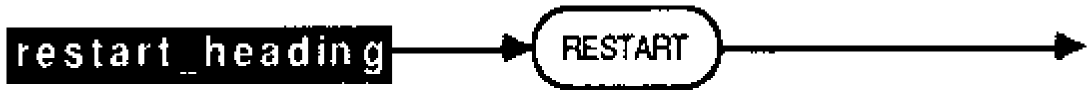
A GOTO that tries to branch across a restart keyword in either direction is a compile-time error.

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.

Example

```
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
```

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 the table "MC parameters accessible by TDC CL" through the table "Timer parameters."

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.

Example

```
SET NN(10) = 100
SET LP(1).SP. 50
IF PF THEN FAIL
IF DI(1008).PV = On THEN SET FL(05) = Off
```

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 the table "Analog input parameters" through the table "Timer parameters."

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 oint in this MFC
EXTERNALB100
    – regulatory point n different MFC
EXTERNALVALVE
    – digital output
EXTERNALSubSeq
    – process module n this MFC
    ...
SET A100.OP = 50
READ NN(10) FROM B100.PV
OPEN VALVE
    – state change NITIATE 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.

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

The table "MC parameters accessible by TDC CL" 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" have parameters, which are listed in “Table 23: Analog input parameters” through to “Table 30: Timer parameters”.

The parameters under the SOPL Name heading marked with an * in “Table 23: Analog input parameters” through to “Table 30: Timer parameters” (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 the table "MC parameters accessible by TDC CL" 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 22: MC parameters accessible by TDC CL

Parameter name	SOPL name	Data type	Array bounds[13]	Description	Remarks
ADFL	PFn	Logical		A/D slot fail	View only
AI	AI	Analog input[14]	1–16(p)	Analog input	p=1–16[15]
AO	AO	Analog output[14]	11–164 (s*10+p)	Analog output	s=1–16 p=1–4[15]
CI	CI	Counter input[14]	11–164 (s*10+p)	Counter input	
DI	DI	Digital input[14]	101–1616 (s*100+p)	Digital input	s=1–16 p=1–16[15]
DO	DO	Digital output[14]	11–168 (s*10+p)	Digital output	s=1–16, p=1–8[15]
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[14]	1–16 (s)	Regulatory ctrl	s=1–16[15]
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[14]	1–32	Timer	
TOG	TG	Logical		Time-out gate	View only

13 **Array index must be an integer constant. If array bounds are not given, the parameter is scalar.**

14 These MFC parameters have parameters; see "Table 23: Analog input parameters" to "Table 30: Timer parameters". The names here match the names described in these tables.

15 s means Slot number and p means Point number. Only valid Point and Slot numbers are allowed.

Table 23: Analog input parameters

Parameter name[16]	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[17]		
SP	SPE	Number	SP (EU)	
SPP	SP		SP (%)	
STATE	ST	Enum[17]	Accum.state	View only
STRTSTOP	RS	Enum[18]	Accum.restart	

Table 24: Analog output parameters

Parameter name[19]	SOPL name	Data type	Description	Remarks
MODATTR		Enum[20]	Mode attribute	View only
MODE	MD	MCMODE[21]	MODE	
NMODATTR		Enum[20]	Normal mode attribute	View only
NMODE		MCMODE[21]	Normal mode	
OP	OV	Number	Output value (%)	
PRTFL	PF	Logical	Partial failure	View only

Table 25: Counter input parameters

Parameter name[22]	SOPL name	Data type	Descriptions	Remarks
AV	AV	Number	Accum.value	View only

---

[16] **The parameters for this table are scalar.**

[17] See the section below titled "Enumeration parameters."

[18] See the section below titled "MODE and NMODE enumeration states."

[19] **The parameters for this table are scalar.**

[20] See the section below titled "Enumeration parameters."

[21] See the section below titled "MODE and NMODE enumeration states."

[22] **The parameters for this table are scalar.**

Parameter name[22]	SOPL name	Data type	Descriptions	Remarks
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[23]		
STATE	ST		Counter state	View only
STRTSTOP	RS		Counter restart	

Table 26: Digital input parameters

Parameter name[24]	SOPL name	Data type	Description	Remarks
PRTFL	PF	Logical	Partial fail	View only
PTINAL	AS		Point in alarm	
PV	ST*		Input state	

Table 27: Digital output parameters

Parameter name[25]	SOPL name	Data type	Description	Remarks
MODATTR		Enum[26]	Mode attribute	View only
MODE	MD	MCMODE[27]	Mode	
NMODATTR		Enm[26]	Normal mode	View only
NMODE		MCMODE[27]	Normal mode	
OP	ST*	Logical	Output state	
PRTFL	PF		Partial failure	View only
PV	FB		Feedback input	

Table 28: Logic block parameters

Parameter name[28]	SOPL name	Data type	Description	Remarks
PV		Logical	Current state	View only

22 **The parameters for this table are scalar.**

23 See the section below titled "Enumeration parameters."

24 **The parameters for this table are scalar.**

25 **The parameters for this table are scalar.**

26 See the section below titled "Enumeration parameters."

27 See the section below titled "MODE and NMODE enumeration states."

28 **The parameters for this table are scalar.**

Table 29: LOOP (Regulatory or Monitoring control slot) parameters

Parameter name[29]	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	
LM	LM	Logical	LM mode	View only
MODATTR		Enum[30]	Mode attribute	
MODE	MD	MCMODE[31]	Mode	
NMODATTR		Enum[30]	Normal mode attr	View only
NMODE		MCMODE[31]	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)	

[29] **The parameters for this table are scalar.**

[30] See the section below titled "Enumeration parameters."

[31] See the section below titled "MODE and NMODE enumeration states."

Parameter name[29]	SOPL name	Data type	Description	Remarks
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	
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	

Table 30: Timer parameters

Parameter name[32]	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*	Enum[33]	Timer state	View only
STRTSTOP	RS		Timer restart	
TIMELEFT	RV	Number	Remaining time	View only
TIMOUTFL	AS	Logical[34]	Time-out flag	

Enumeration parameters

Enumeration parameters have the following data types:

Parameter Name	Enumeration Name	Enumeration States
MODATTR, NMODATTR	MODATTR	OPERATOR/PROGRAM

---

[29] **The parameters for this table are scalar.**

[32] **The parameters for this table are scalar.**

[33] See the section below titled "Enumeration parameters."

[34] See the section below titled "Accessing MFC parameters."

Parameter Name	Enumeration Name	Enumeration States
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.

MODE and NMODE enumeration states

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/PMAN/PAUTO/PCAS/PCOM/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 cannot 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.

Accessing MFC parameters

The LCN view of the parameter T1MOUTFL 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
```

Related topics

“Creating CL programs” on page 76

User-written subroutines

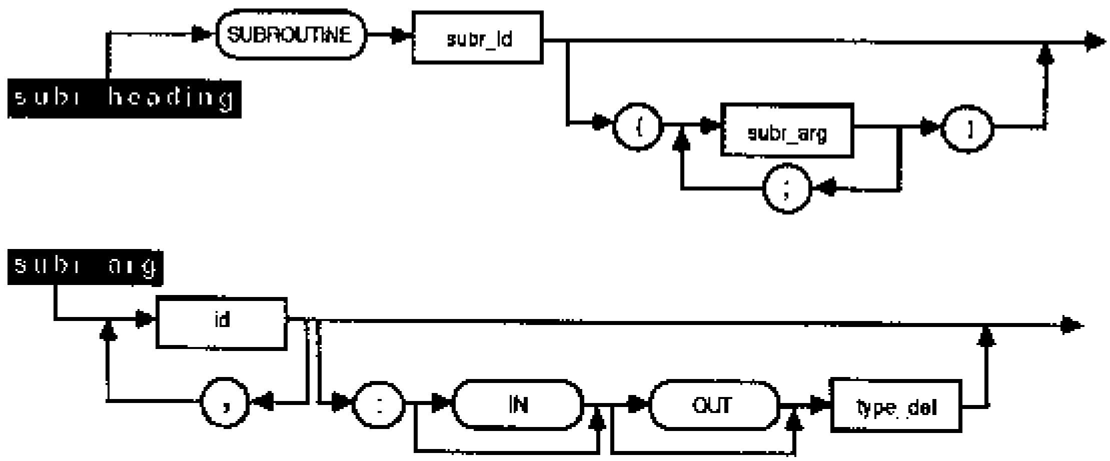
This section describes user-written TDC CL subroutines. A program can call subroutines that are user-written or system-supplied: user-written subroutines must use the facility described here, but system-supplied subroutines need not be written in TDC CL.

User-written TDC CL subroutines are referred to as Sequence subroutines.

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 the table "Analog input parameters" through the table "LOOP (Regulatory or Monitoring control slot) parameters."

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.

Example

```
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.

Example

```
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 (see the topic titled "Notation used for syntax production rules") are included within more complex diagrams.

Related topics

"Syntax diagram summary" on page 139
"Notation used for syntax production rules" on page 146
"Production rules" on page 147
"Control Language basics" on page 82
"Notation used for syntax production rules" on page 146

Syntax diagram summary

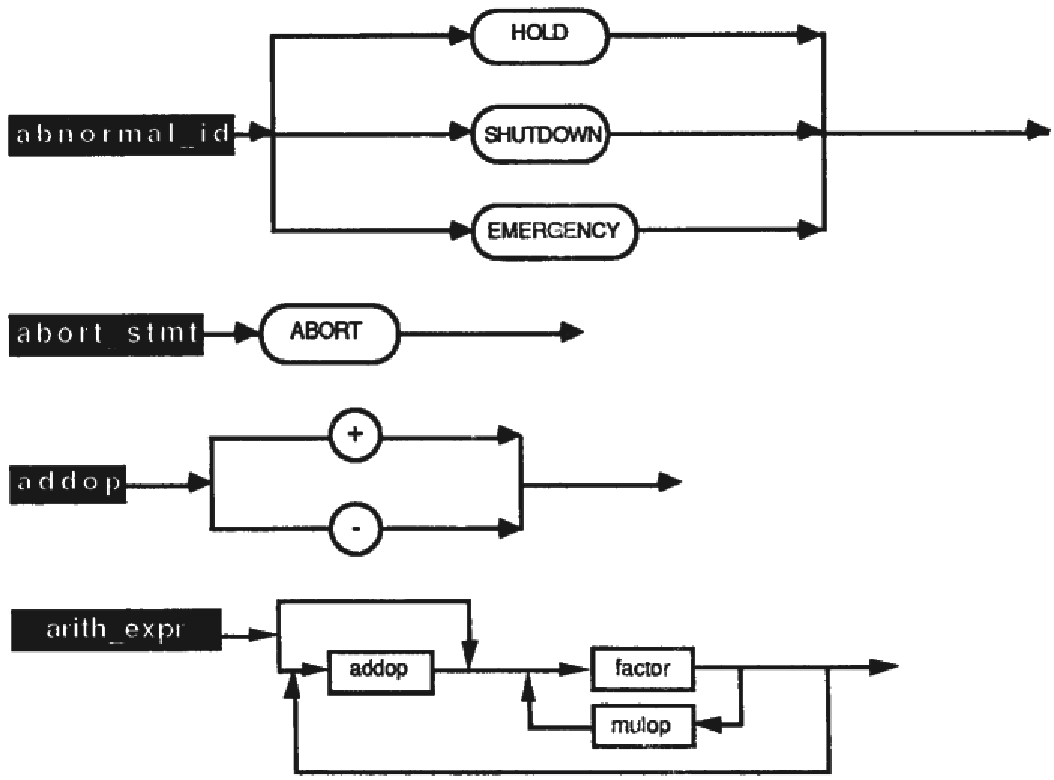
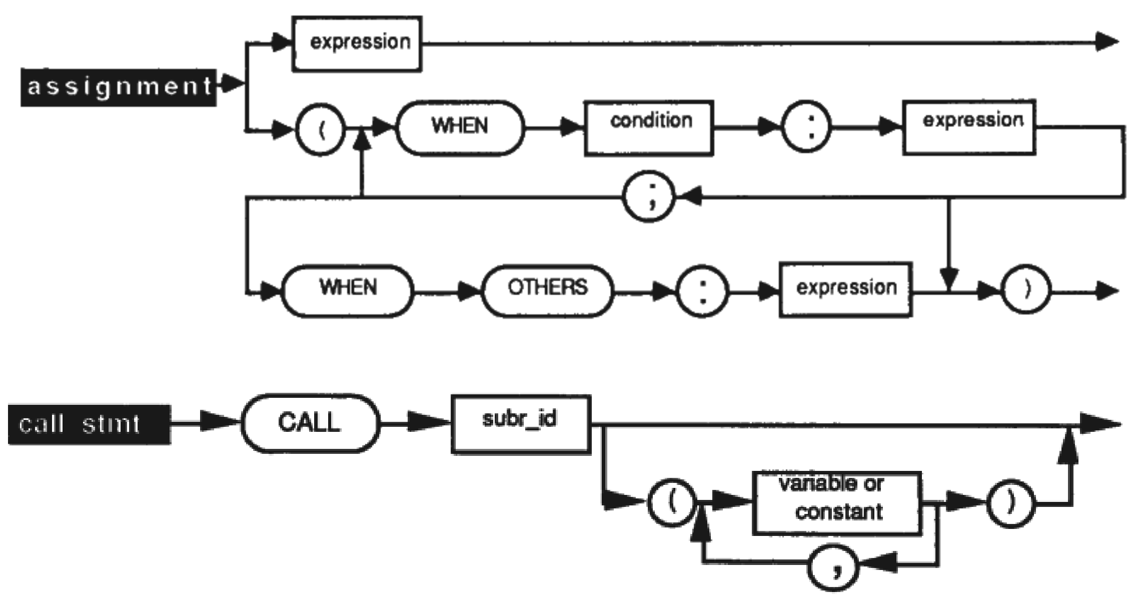
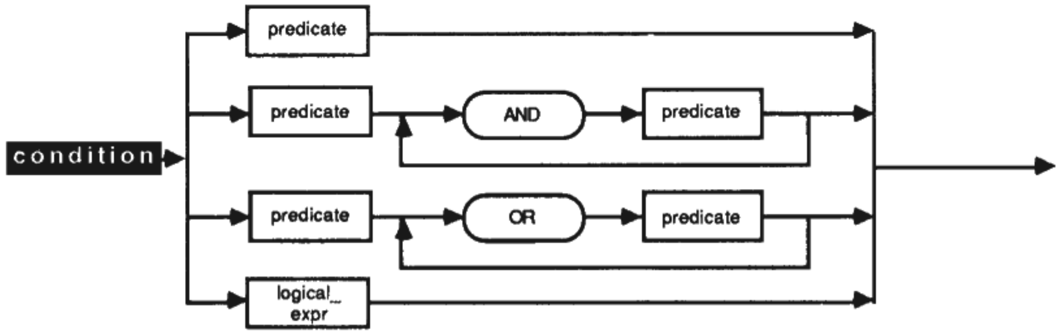
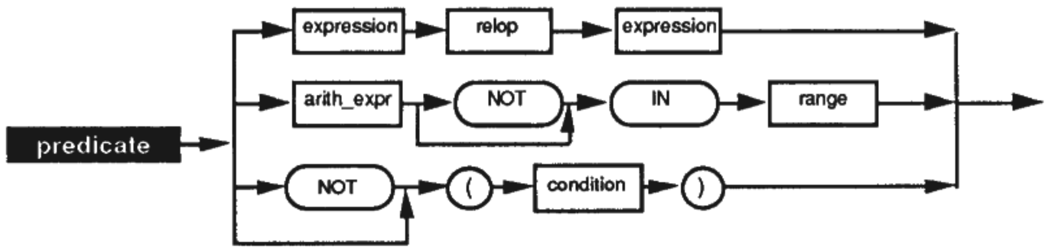
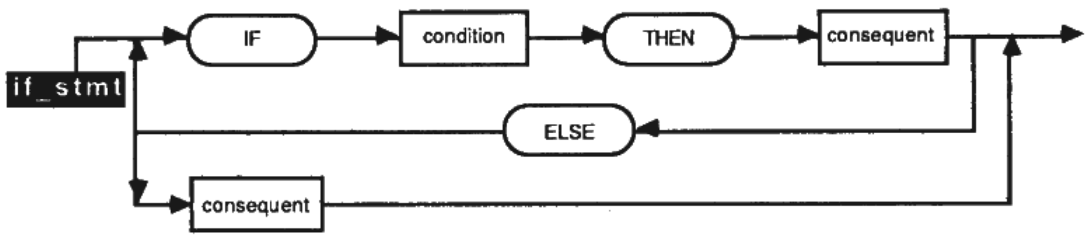


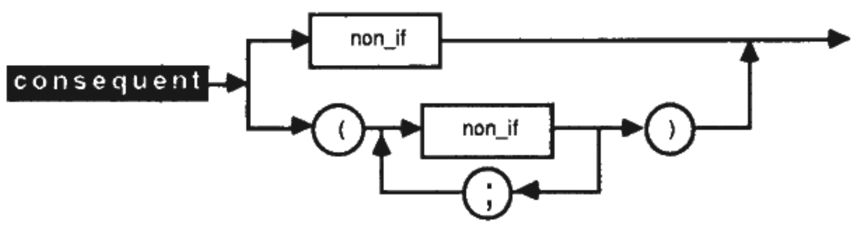
Figure 20: Abnormal_id - Arith_expr




condition
predicate
predicate
AND
predicate
predicate
OR
predicate
logical_
expr


predicate
expression
relop
expression
arith_expr
NOT
IN
range
NOT
(
condition
)


if_stmt
IF
condition
THEN
consequent
ELSE
consequent


consequent
non_if
(
non_if
)
;

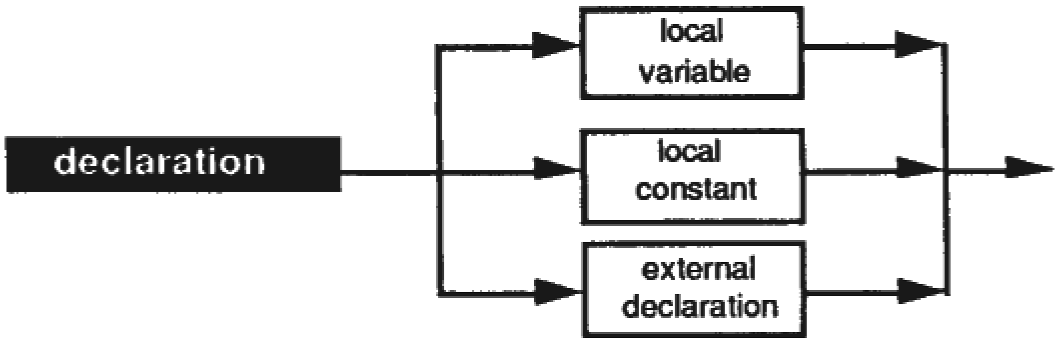


Figure 21: \assignment - Declaration

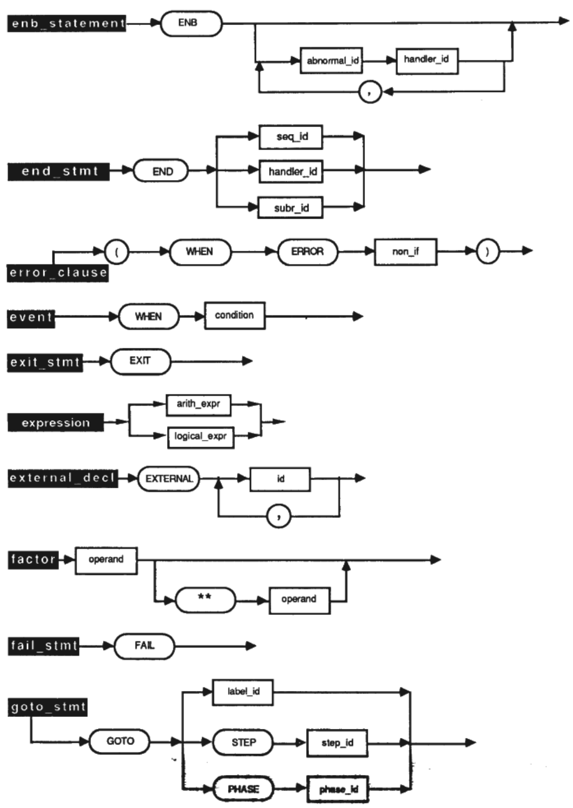


Figure 22: End_statement - Goto_stmt

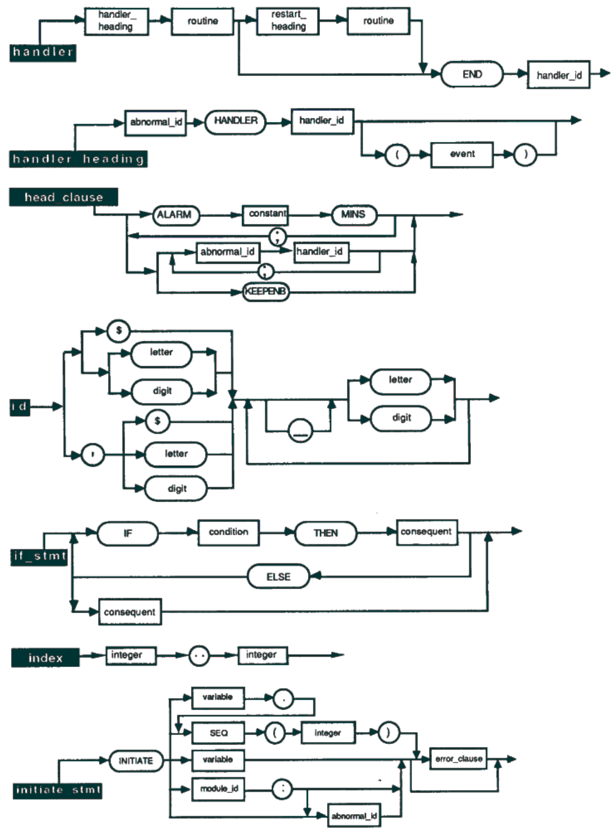


Figure 23: Handler - Initiate_stmt

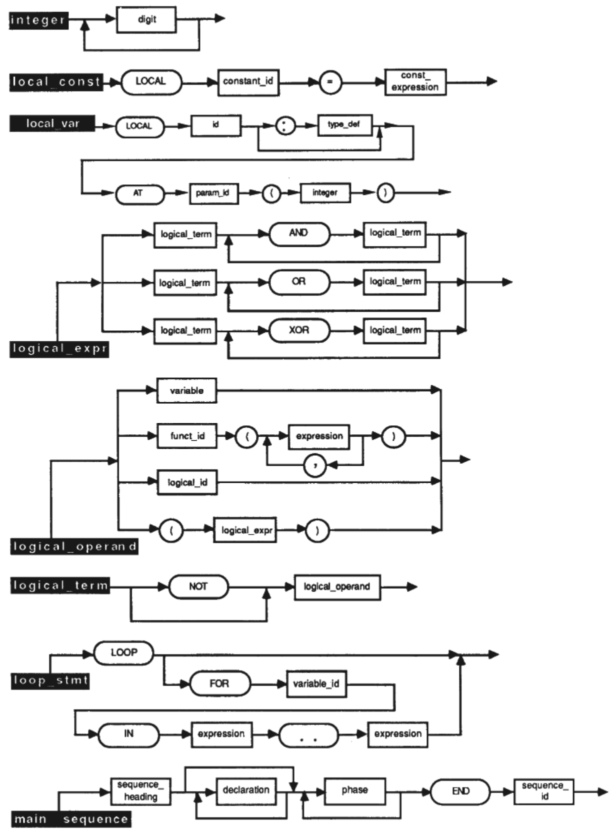


Figure 24: Integer - Main_sequence

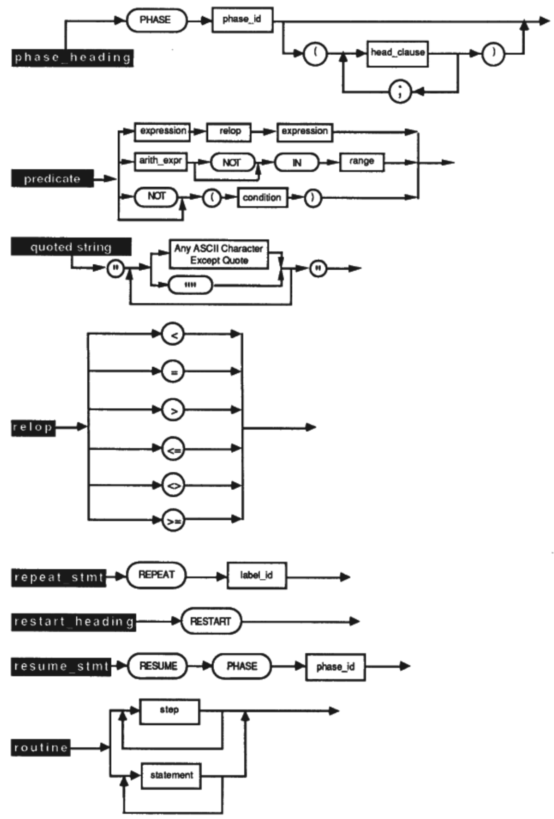


Figure 25: Phase_heading - Routine

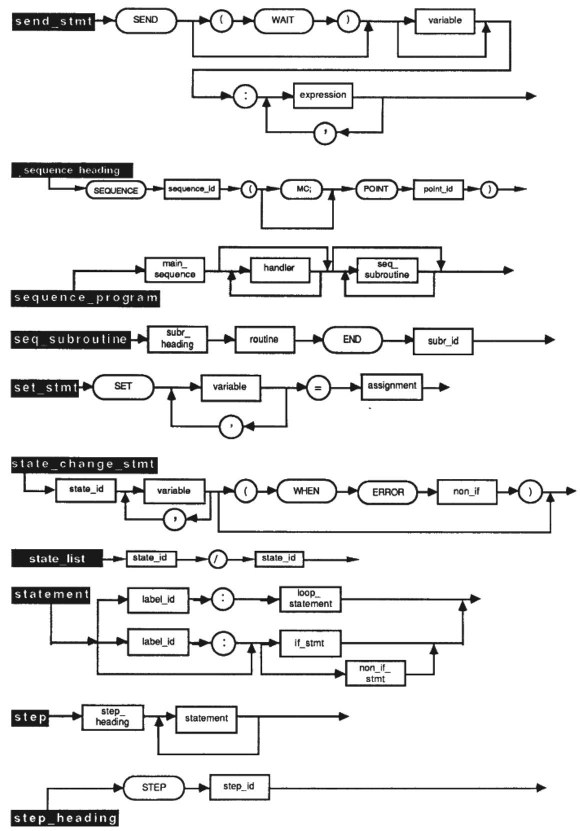


Figure 26: Send_stmt - Step_heading

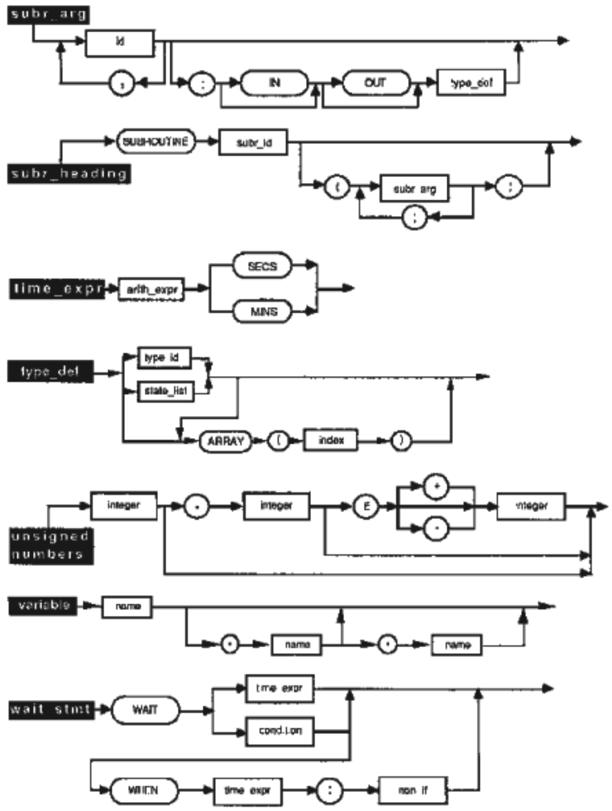


Figure 27: Sugr_arg - Wait_stmt

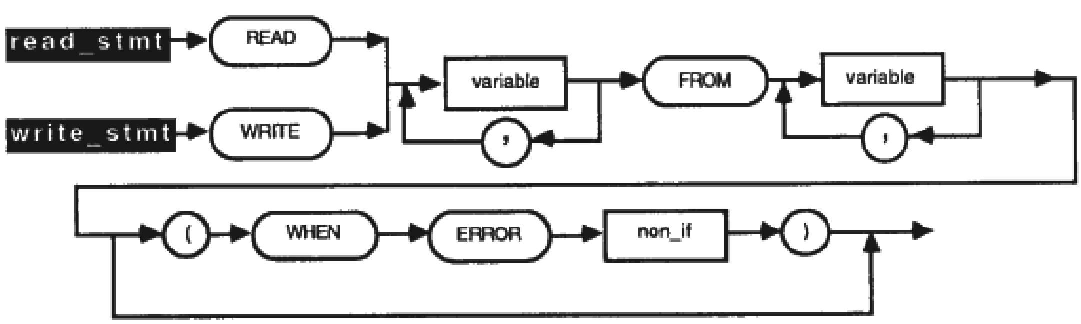


Figure 28: Read_stmt - Write_stmt

Notation used for syntax production rules

The following presentation of TDC CL syntax effectively follows BNF notation conventions as follows:

Sequences of lowercase 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 I, 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 set point, 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).

Related topics

"Syntax summary" on page 138

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]
```

Troubleshooting Honeywell TDC 3000 Data Hiway issues

The following is a summary of some of the applications and utility programs that can be used to troubleshoot the system; see the Station Help menu item for more information.

Application/Utility	Used For	Where to find it	Reference
alglst	Lists the free algorithm blocks numbers	Type the command into a Command Prompt Window on the primary server	Station Help
DHEB Monitor	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
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
display	Used to show the description for server error codes	Type the command into a Command Prompt Window on the primary server	Station Help
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
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	See the topic titled "Configuration monitoring of the Honeywell TDC Data Hiway box" in this document
lisscn	Used to analyze and optimize the scanning load	Type the command into a Command Prompt Window on the primary server	Station Help
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
prpmon	Used to monitor the point processing performance	Type the command into a Command Prompt Window on the primary server	Not applicable
rtusum	Lists all points assigned to a particular controller	Type the command into a Command Prompt Window on the primary server	Station Help
Station Display Page 10 (Channel Scanning Statistics display)	Used to monitor the load on the scanning subsystem	Select Request Page icon or press F5 and enter the page number 10	Station Help

Application/Utility	Used For	Where to find it	Reference
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
tdc_cl	Used to compile TDC CL/MC sequence programs	Type the command into a Command Prompt Window on the primary server	See the topic titled "Compiling CL programs" in this document
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	See the topic titled "Testing Honeywell TDC Data Hiway communications with the server" in this document
trace	Used to record communications activity for Stations and channels.	Type the command into a Command Prompt Window on the primary server	Station Help

Related topics

"Testing Honeywell TDC 3000 Data Hiway communications with the server" on page 153

"Monitoring the system status of Honeywell TDC 3000 Data Hiway" on page 154

"Configuration monitoring of the Honeywell TDC Data Hiway box" on page 156

"Error codes for Honeywell TDC 3000 Data Hiway" on page 160

"Planning considerations for installing and configuring Honeywell TDC 3000 Data Hiway controllers" on page 7

"Compiling CL programs" on page 77

"Configuration monitoring of the Honeywell TDC Data Hiway box" on page 156

"Testing Honeywell TDC 3000 Data Hiway communications with the server" on page 153

Testing Honeywell TDC 3000 Data Hiway communications with the server

You use the Honeywell TDC 3000 Data Hiway test utility, **tdctst**, to test communications between the server and the TDC 3000 Data Hiway controller after you have downloaded channel and controller definitions to the server database.

Prerequisites

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

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 the tdctst utility

1. Open a Command Prompt window.
2. Type **tdctst** and then press Enter.
3. When prompted, type the channel number of the Data Hiway, followed by `find`. For example:

   ```
   CHN01
   find
   ```

   To test the redundant communication channel, use `CHN01D`.

   If this is a new controller 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.

   A list of addresses and box types on the Hiway appears.
4. Check that the addresses correspond to the physical connections on the Data Hiway and to the server controller definitions.
5. Type `q` to quit.

Related topics

"Planning considerations for installing and configuring Honeywell TDC 3000 Data Hiway controllers" on page 7
"Troubleshooting Honeywell TDC 3000 Data Hiway issues" on page 151

Monitoring the system status of Honeywell TDC 3000 Data Hiway

Attention
The indicators on the SCADA Controllers display represent the communication status, as well as the TDC box status and box error descriptions.

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 failover	Temporary loss of process view and control.	The server failover 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 failover 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 nn* command or HTD switch panel.
A80 Controller COMMS alarm	Recovery from power loss	Partial loss of process view and control.	Use CLR nn command for PIU, DHP and HIM devices. Use *STR nn* 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 nn* command for PIU, DHP and HIM devices. Use *STR nn* 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 nn* 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.

Symptom	Possible Causes	Consequences	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.

Configuration monitoring of the Honeywell TDC Data Hiway box

Configuration monitoring reads up TDC box resident information and compares it with the server image of the box. Parameters that 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 SCADA Controllers 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.

CB box configuration parameters

Table 31: CB box configuration parameters

Tracked on change	Alarmed on change
Bias	Alarm type
Decimal place	Algorithm
Deviation HI limit[35]	EU conversion
Deviation LO limit[35]	Override algorithm parameters
EU span[35]	Past mode recall
EU zero[35]	PV initialize
Integral HI limit	PV tracking
Integral LO limit	Square root indication
K (gain, multiplier)	Temperature units
OP HI limit	X - input
OP LO limit	Y - input
PV HI limit[35]	
PV LO limit[35]	
Ratio	
RV (scale, bias)	
T (integral, lag, multiplier)	
T2 (derivative, lead, bias)	
TD	

EC box configuration parameters

Table 32: EC box configuration parameters

Tracked on change	Alarmed on change
Bias	Alarm type
Deviation HI limit[36]	Algorithm
Deviation LO limit[36]	Control
EU span[36]	Equation

[35] Tracked into and out of the server point parameter (if built).

[36] Tracked into and out of the server point parameter (if built). EC SP range not supported; SP will take Range of PV.

Tracked on change	Alarmed on change
EU zero[36]	Indication
Integral HI limit	M - input
Integral LO limit	Past mode recall
K (gain, multiplier)	PV initialize
K1 (integral, lag, multiplier)	PV tracking
K2 (derivative, lead, bias)	Square root
OP HI limit	TOG interval
OP LO limit	X - input
PV HI HI limit[36]	Y conversion
PV HI limit[36]	Y - input
PV LO limit[36]	Z - input
PV LO LO limit[36]	
PV rate limit[36]	
Ratio	
X - decimal place	
Y - decimal place	

MFC and AMC box configuration parameters

Attention

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 must therefore be taken when viewing box configuration data while box is in IDLE state.

Table 33: MFC and AMC box configuration parameters

Tracked on change	Alarmed on change
Decimal place[37]	Alarm type
Deviation HI limit[37]	Algorithm
Deviation LO limit[37]	Board type
EU span[37]	EU conversion
EU zero[37]	Indication
Integral HI limit	Input override
K (gain, multiplier)	Logic blocks
K1 (integral, lag, multiplier)	Override control algo
K2 (derivative, lead, bias)	Override equation algo
OP HI limit	Past mode recall
OP LO limit	Pulse width (DO)
PV HI limit[37]	PV initialize
PV LO limit[37]	PV tracking
Rate HI limit	Sensor type
Rate LO limit	Square root

[37] Tracked into and out of the server point parameter (if built).

Tracked on change	Alarmed on change
Ratio	Temperature units
RV (scale, bias)	X - input
SP HI limit	Y - input
SP LO limit	
TD (digital filter)	

PIU box configuration parameters

Table 34: PIU box configuration parameters

Tracked on change	Alarmed on change
Decimal place[38]	Alarm type
Deviation HI limit[38]	Board type[39]
Deviation LO limit[38]	Notify suppress
EU span[38]	Open T/C detect
EU zero[38]	Pulse width
PV HI limit[38]	Scan frequency
PV LO limit[38]	Sensor type
	Significant change
	Smoothing
	Square root
	Temperature units

DHP box configuration parameters

Table 35: DHP box configuration parameters

Tracked on change	Alarmed on change
Actual scan time	Address[40]
Decimal place[41]	Alarm type
Deviation HI limit[41]	Board type
Deviation LO limit[41]	Configured scan time[40]
EU span[41]	Device[40]
EU zero[41]	Device address[40]
PV HI limit[41]	Hot start[40]
PV LO limit[41]	Keep alive address[40]
	Model type[40]
	Notify suppress

[38] Tracked into and out of the server point record parameter (if built). The appropriate slot change flag is also issued by the server.

[39] Reset TDC box on data entry.

[40] Reset TDC box on data entry.

[41] Tracked into and out of the server point record parameter (if built).

Tracked on change	Alarmed on change
	Port[40]
	Protocol[40]
	Range[40]
	Sensor type
	Significant change
	Square root

Related topics

"I/O slot definitions for Honeywell TDC 3000 Data Hiway" on page 40
"Troubleshooting Honeywell TDC 3000 Data Hiway issues" on page 151

Error codes for Honeywell TDC 3000 Data Hiway

The following error codes appear in the alarm messages and on the SCADA Controllers display. TDC box status and error descriptions also appear on the SCADA Controllers displays.

Related topics

"Hiway addressing/communication error codes" on page 160
"HLPIU (3xx), LLPIU (4xx), and LEPIU (5xx) error codes" on page 163
"DHP or 620 HIM error codes" on page 165
"Basic controller primary (1xx) and reserve (6xx) error codes" on page 167
"Extended controller primary (7xx) and reserve (8xx) error codes" on page 169
"Multifunction controller (MFC and AMC) error codes" on page 171
"UAC primary controller codes" on page 176
"UAC Reserve Controller Director CB (6xx) and EC (8xx) error codes" on page 177
"UAC Reserve Controller Director MC error codes" on page 178
"UAC reserve controller CB (6xx) and EC (8xx) error codes reported by the RCD" on page 181
"UAC reserve controller MC error codes reported by the RCD" on page 183

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 preempted 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 preempted 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	

Error	H11
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	
Solution	No zero offset error.

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 call up.

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.

Error	1102
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	Undefined.
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.

Error	n09
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.

Error	n07
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 n38
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.

Error	921
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.

Error	944
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 SCADA Controllers 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).

Error	S1xx
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 the 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 overrun.
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.

Error	1055
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.

Error	**878**
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.

Error	**1069**
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	1086
State	RC Fail.
Description	UAC I/F card failure.
Solution	Reserve Controller Director (RCD) has failed. See the *TDC service manual*.

Notices

Trademarks

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

OneWireless™ is a trademark of Honeywell International, Inc.

Other trademarks

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

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

Third-party licenses

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

Documentation feedback

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

http://www.honeywellprocess.com/support

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

hpsdocs@honeywell.com

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

How to report a security vulnerability

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

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

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

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

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

- Send an email to security@honeywell.com.

  or

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

Support

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

Training classes

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

Index

A

abnormal condition handlers 123
accessing parameters 126
address syntax 58–60, 64–68
addressing
  Honeywell TDC 3000 Data Hiway controllers 58–60, 64–68
advanced multifunction controller 12
alglst utility 151
AMC error codes 171
architectures
  Honeywell TDC 3000 Data Hiway 16, 17

B

basic controller 12
Basic Controller (CB) addressing 58
Basic Controller (CB) command 43
basic controller error codes 167
baud 24, 34, 36
bckld utility 151
box address 38
box type 38
built-in functions and subroutines 136

C

C Link addresses 73
channels
  Honeywell TDC 3000 Data Hiway channel 32
  Honeywell TDC 3000 Data Hiway controllers 31
  reference 31
checksum 34, 36
CL acronym libraries 72
CL Acronym Library Number 38
CL program directories 76
CL programs 76
CL sequence messages 77
CL software environment 75
commands and utilities
  alglst 151
  Basic Controller (CB) 43
  bckld 151
  Data Hiway switch 43
  DHEB Monitor 151
  DHP commands 44
  Diagnostic Capture Tool (DCT) 151
  Extended Controller (EC) 43
  hdwbckbld 151
  hdwbld 151
  hdwconfig 40, 47, 151
  HIM commands 44
  Honeywell TDC 3000 Data Hiway controllers 43, 44
  lisscn 151
  listag 151
  Multifunction Controller (MC) commands 43
  PIU commands 44
  pntbld 151
  pntdel 151
  prpmon 151
  Reserve Controller Director (RCD) commands 44
  rtusum 151
  tdc_cl 151
  tdctst 151, 153
  trace 151
communications
    testing
      Honeywell TDC 3000 Data Hiway controllers 153
compile time errors 78
compiling CL programs 77
configuration tasks 72
configuring
  Honeywell TDC 3000 Data Hiway controllers 7
connect timeout 32
Control Language reference 71, 72, 75, 82, 102, 120, 138
controllers
    configuring
      Honeywell TDC 3000 Data Hiway controllers 7
  Honeywell TDC 3000 Data Hiway controllers 7, 9, 10, 12, 13, 15–20, 22–25, 27, 29, 31, 34, 36–38, 40–44, 46, 47, 49, 51, 53, 55, 58–60, 64–68, 71–73, 75–79, 82, 90, 93, 96, 102, 119, 120, 123, 125, 126, 134, 136, 138, 139, 146, 147, 151, 153, 154, 156, 160, 163, 165, 167, 169, 171, 176–178, 181, 183
  reference 31

D

Data Hiway Bridge 17, 23–25, 27, 29
Data Hiway Ethernet Bridge 16, 18–20, 22, 24
Data Hiway port 12
Data Hiway Port (DHP) addressing 65
Data Hiway switch command 43
data types 90
devices supported 12
DHB 17, 23–25, 27, 29
DHEB 16, 18–20, 22, 24
DHEB Monitor 151
DHP commands 44
DHP error codes 165
Diagnostic Capture Tool (DCT) 151

diagnostics
Honeywell TDC 3000 Data Hiway controllers 153
documentation
Honeywell TDC 3000 Data Hiway 9
download permitted 37
downloading
sequence programs 78

E

embedded compiler directives 119
error codes
AMC error codes 171
basic controller error codes 167
DHP error codes 165
extended controller error codes 169
HIM error codes 165
Hiway addressing/communication 160
HLPIU (3xx) 163
Honeywell TDC 3000 Data Hiway controllers 160, 163, 165, 167, 169, 171, 176–178, 181, 183
LEPIU (5xx) 163
LLPIU (4xx) 163
MFC error codes 171
multifunction controller error codes 171
primary (1xx) error codes 167
primary (7xx) error codes 169
primary controller error codes 176
reserve (6xx) error codes 167
reserve (8xx) error codes 169
reserve controller error codes 177, 178, 181, 183
errors
compile time errors 78
run time errors 79
exception scanning 53
extended controller 12
Extended Controller (EC) addressing 59
Extended Controller (EC) command 43
extended controller error codes 169

F

fail alarm limit 32, 38
fatal errors 78, 79
Fault Tolerant Ethernet (FTE)
Honeywell TDC 3000 Data Hiway 17
FTE (Fault Tolerant Ethernet)
Honeywell TDC 3000 Data Hiway 17

G

generic hardware addressing 66

H

hdwbckbld utility 151
hdwbld utility 151
hdwconfig utility 40, 47, 151
HIM commands 44
HIM error codes 165
Hiway address 24
Hiway owner 37
HLPIU (3xx) error codes 163
Honeywell TDC 3000 Data Hiway controllers
abnormal condition handlers 123
accessing parameters 126
additional tasks 42
addressing 58–60, 64–68
advanced multifunction controller 12
AMC error codes 171
architectures 16, 17
backing up box configuration 47
basic controller 12
Basic Controller (CB) addressing 58
Basic Controller (CB) command 43
basic controller error codes 167
baud 24, 34, 36
box address 38
box type 38
built-in functions and subroutines 136
C Link addresses 73
channels 31, 32, 34, 36, 37
checksum 34, 36
CL acronym libraries 72
CL Acronym Library Number 38
CL program directories 76
CL programs 76
CL sequence messages 77
CL software environment 75
commands 43, 44
comparisons 13
compile time errors 78
compiling CL programs 77
configuration management 49
configuration tasks 72, 73
configuring 7
connect timeout 32
considerations for operation 15
Control Language reference 71, 72, 75, 82, 102, 120, 138
controllers 31, 38
Data Hiway Bridge 17, 23–25, 27, 29
Data Hiway Ethernet Bridge 16, 18–20, 22, 24
Data Hiway port 12
Data Hiway Port (DHP) addressing 65
Data Hiway switch command 43
data types 90
devices supported 12
DHB 17, 23–25, 27, 29
DHEB 16, 18–20, 22, 24
DHP commands 44
DHP error codes 165
documentation 9
download permitted 37
downloading sequence programs 78
embedded compiler directives 119
errors 78, 79
exception scanning 53
extended controller 12
Extended Controller (EC) addressing 59
Extended Controller (EC) command 43

extended controller error codes 169
fail alarm limit 32, 38
fatal errors 78
Fault Tolerant Ethernet (FTE) 17
FTE (Fault Tolerant Ethernet) 17
functions 136
generic hardware addressing 66
getting started 7
hdwconfig utility 47
HIM commands 44
HIM error codes 165
Hiway address 24
Hiway owner 37
HLPIU (3xx) error codes 163
I/O slot definitions 40
idle timeout 34, 36
language expressions and conditions 96
language rules and elements 82
language variables and declarations 93
LCS620 HIM addressing 65
LCS620 Hiway interface module 12
LEPIU (5xx) error codes 163
linearization algorithms 67
LLPIU (4xx) error codes 163
logic blocks 73
marginal alarm limit 32, 38
MFC error codes 171
monitoring configuration 156
monitoring system status 154
multifunction controller 12
Multifunction Controller (MC) commands 43
Multifunction Controller (MFC and AMC) addressing 60
multifunction controller error codes 171
notation for syntax production rules 146
number of data bits 34, 36
operational considerations 15
optimizing scanning performance 41
parameters 126
parity 24, 34, 36
phase names 72
PIU commands 44
PIU/DHP significant change reporting 37
point displays 68
points 51, 55
primary (1xx) error codes 167
primary (7xx) error codes 169
primary controller error codes 176
Process Interface Unit (PIU addressing 64
process interface units 12
process module points 72
production rules 147
program development 75–79
program statement definitions 102
protocol 24
read timeout 32
reference 71, 72, 75, 82, 102, 120, 138
refresh enabled for computer mode timeout 37
reserve (6xx) error codes 167
reserve (8xx) error codes 169
reserve basic controller director 12
Reserve Controller Director (RCD) commands 44
reserve controller error codes 177, 178, 181, 183
reserve extended controller director 12
reserve multifunction controller director 12
restart routines 125
restoring box configuration 47
RS-232 24, 25, 34, 36
RS-422 24, 25
RS-485 34, 36
run time errors 79
scanning 53
scanning performance 41
send string item 72
sequence names 72
sequence programs 46, 78, 79, 120
serial port properties 34, 36
statements 102, 119
step labels 72
stop bits 34, 36
structures 120, 123, 125, 126, 134, 136
subroutines 136
supported devices 12
syntax 138, 139, 146, 147
TCP host name 34, 36
TCP port number 34, 36
tdctst 153
terminal server port properties 34, 36
terms 10
testing communications 153
TPS CL MC compiler 79
troubleshooting 22, 29, 79, 151, 153, 154, 156, 160, 163, 165, 167, 169, 171, 176–178, 181, 183
unexpanded PIU 38
upload permitted 37
uploading sequence programs 79
user-written subroutines 134
warnings 78
XON/XOFF 34, 36

I

I/O slot definitions 40
idle timeout 34, 36

L

LCS620 HIM addressing 65
LCS620 Hiway interface module 12
LEPIU (5xx) error codes 163
linearization algorithms 67
lisscn utility 151
listag utility 151
LLPIU (4xx) error codes 163
logic blocks 73

M

Main tab
Honeywell TDC 3000 Data Hiway channel 32

Honeywell TDC 3000 Data Hiway controller 38
marginal alarm limit 32, 38
MFC error codes 171
monitoring
configuration 156
system status 154
multifunction controller 12
Multifunction Controller (MC) commands 43
Multifunction Controller (MFC and AMC) addressing 60
multifunction controller error codes 171

N

notation for syntax production rules 146
number of data bits 34, 36

P

parameters 126
parity 24, 34, 36
phase names 72
PIU commands 44
PIU/DHP significant change reporting 37
pntbld utility 151
pntdel utility 151
points
Honeywell TDC 3000 Data Hiway controllers 51, 55
reference 51, 55
port properties
configuring 34, 36
Honeywell TDC 3000 Data Hiway channel 34, 36
primary (1xx) error codes 167
primary (7xx) error codes 169
primary controller error codes 176
Process Interface Unit (PIU addressing 64
process interface units 12
process module points 72
production rules 147
program development 75–79
program statement definitions 102
protocol 24
prpmon utility 151

R

read timeout 32
refresh enabled for computer mode timeout 37
reserve (6xx) error codes 167
reserve (8xx) error codes 169
reserve basic controller director 12
Reserve Controller Director (RCD) commands 44
reserve controller error codes 177, 178, 181, 183
reserve extended controller director 12
reserve multifunction controller director 12
restart routines 125
RS-232
connections 24, 25, 34, 36
RS-422
connections 24, 25
RS-485
connections 34, 36
rtusum utility 151
rules
production rules 147
run time errors 79

S

scanning
optimizing performance 41
send string item 72
sequence names 72
sequence programs 46, 78, 79, 120
serial port properties 34, 36
statements 102, 119
step labels 72
stop bits 34, 36
structures 120, 123, 125, 126, 134, 136
supported devices 12
syntax
TDC CL syntax 138, 139, 146, 147

T

TCP host name 34, 36
TCP port number 34, 36
TDC 3000 controllers
, See Honeywell TDC 3000 Data Hiway controllers
tdc_cl utility 151
tdctst utility 151, 153
terminal servers
port properties 34, 36
testing communications
Honeywell TDC 3000 Data Hiway controllers 153
TPS CL MC compiler 79
trace utility 151
troubleshooting
Honeywell TDC 3000 Data Hiway controllers 22, 29, 151, 154, 156, 160, 163, 165, 167, 169, 171, 176–178, 181, 183
sequence programs 79

U

unexpanded PIU 38
upload permitted 37
uploading
sequence programs 79
user-written subroutines 134
utilities
alglst 151
bckld 151
DHEB Monitor 151
Diagnostic Capture Tool (DCT) 151
hdwbckbld 151
hdwbld 151
hdwconfig 40, 47, 151
lisscn 151
listag 151
pntbld 151

pntdel 151
prpmon 151
rtusum 151
tdc_cl 151
tdctst 151, 153
trace 151

W

warnings
compile time errors 78

X

XON/XOFF 34, 36