

# Industrial<sup>IT</sup> 800xA - System 800xA for MOD 300 System Version 4.1

## Operation





**Industrial<sup>IT</sup>**  
**800xA - System**  
**800xA for MOD 300**  
**System Version 4.1**

**Operation**

---

## NOTICE

The information in this document is subject to change without notice and should not be construed as a commitment by ABB. ABB assumes no responsibility for any errors that may appear in this document.

In no event shall ABB be liable for direct, indirect, special, incidental or consequential damages of any nature or kind arising from the use of this document, nor shall ABB be liable for incidental or consequential damages arising from use of any software or hardware described in this document.

This document and parts thereof must not be reproduced or copied without written permission from ABB, and the contents thereof must not be imparted to a third party nor used for any unauthorized purpose.

The software or hardware described in this document is furnished under a license and may be used, copied, or disclosed only in accordance with the terms of such license.

This product meets the requirements specified in EMC Directive 89/336/EEC and in Low Voltage Directive 72/23/EEC.

Copyright © 2004 by ABB.  
All rights reserved.

Release: May 2005  
Document number: 3BUR002418R4101

## TRADEMARKS

Registrations and trademarks used in this document include:

Acrobat	Registered trademark of Adobe Systems, Incorporated.
Adobe	Registered trademark of Adobe Systems, Incorporated.
Advant	Registered trademark of ABB Inc.
AdvaBuild	Registered trademark of ABB Inc.
Industrial IT	Trademark of ABB Inc.
Microsoft	Registered trademark of Microsoft Corporation.
Windows	Registered trademark of Microsoft Corporation

---

# TABLE OF CONTENTS

## About This Book

General .....	17
Use of Warning, Caution, Information, and Tip .....	17
Document Conventions .....	18
Terminology .....	19
Related Documentation .....	21
On-line Documentation.....	23
On-line Help.....	23

## Section 1 - Introduction

Document Overview .....	25
Intended User .....	25
Where to Start .....	25
Operation .....	26
Diagnostic Displays .....	26
I/O Displays .....	26
Operation Examples.....	26
On-line Help.....	26
Product Overview .....	27

## Section 2 - Getting Started

General .....	29
Start-up Procedure .....	29
Product Verification .....	29
Log on as a Different User .....	30
Operating Overview .....	31

Accessing Displays .....	31
Accessing Displays Using Browser .....	31
Accessing Displays Using Graphic Displays.....	32
Accessing Displays Using Context Menus .....	32
MOD 300 Objects and Associated Aspects .....	33
Availability of Displays as Indicated by Server Status.....	35
Printing Displays .....	36
Alarm/Event Handling .....	36
<b>Section 3 - CCF Displays</b>	
Operational Displays for Configurable Control Functions .....	37
Loop Detail Displays.....	37
Control and PID Loops .....	38
Device Loops .....	42
Continuous Loops .....	44
Loop Faceplates .....	45
Tag Identification and Indicators .....	45
Process Bar Graphs and Values .....	46
Operator Controls .....	47
Tool Tips .....	50
Loop FCM Display .....	51
Operating Procedures for CCF Displays.....	52
Viewing Trend Data.....	52
Taking Control of a Loop .....	54
Changing Control Parameters.....	54
Tuning/Engineering Procedures for CCF Displays.....	57
Tuning Via the Loop Detail Display.....	57
Engineer Tuning Displays for a Continuous Loop.....	59
Loop FCM Display .....	59
Runtime Versions of Loop Definition and FCM Templates .....	62
Runtime Version of Breakpoint Sets Template.....	62
Engineer Tuning Displays for a Device Loop .....	63
Alarms .....	64

---

Alarm Post and Alarm Check .....	64
Alarm Detection .....	64
Alarm Indication and Viewing.....	65
Alarm Priority .....	65
Alarm Limits and Limiting .....	66
Alarm Acknowledgement .....	67
MOD Group Displays.....	67
Group Status Display .....	68
PID and A/M Status Blocks .....	68
Continuous Loop Status Blocks .....	69
Device Loop Status Blocks .....	70
Indicator Status Blocks .....	70
Alarm Indication and Viewing.....	70
Group Trend Display.....	71
Group Trend Graph.....	71
Group Trend Status Blocks .....	72
Alarm Indication and Viewing.....	72
Group Graphic Display .....	73
Group Alarm Display .....	74
MOD Area Displays .....	75
Area Status Display.....	75
Area Alarm Display .....	77
Area Graphic Display.....	78

## **Section 4 - TCL Displays**

Operational Displays for TCL .....	79
Displays for Sequence and Recipe Development.....	79
TCL Version Mismatch Warning .....	80
TCL / Unit Message Display .....	80
TCL Message Display Format .....	81
Acknowledge and Respond to a Message.....	81
Unit Overview Display .....	82
Unit Detail Display .....	83

Unit Detail Display Format.....	84
Loading the Sequence from the Unit Detail Display .....	84
Loading a Recipe from the Unit Detail Display .....	85
Changing Sequence State, Mode, and Status.....	85
Removing one or more Sequences.....	86
Selecting a Specific Sequence for SFC, Sequence Detail, or Debug Display .....	86
Sequence Detail Display .....	87
Sequence Detail Display Format .....	88
Choosing a Sequence for Sequence Detail .....	88
Changing Sequence State, Mode, and Status.....	88
Manipulating Sequence Steps.....	88
SFC (Sequential Function Chart) Display.....	89
SFC Display Format .....	90
SFC Display Operation .....	91
Sequence Debug Display .....	93
Sequence Debug Display Format .....	93
Choosing a Sequence for Debug Display .....	94
Sequence Source .....	94
Changing Sequence State, Mode, and Status.....	94
Sequence Debug Program Functions .....	94
Find .....	95
Goto .....	95
Trace .....	95
Breakpoint .....	96
Recipe Detail .....	98
Recipe Detail Display Format.....	98
Changing Recipe Values.....	99
TCL Array Plot Display .....	99
TCL Array Plot Format.....	99
Using TCL Array Plot.....	99
Context Menu .....	100

## Section 5 - TLL Displays



Operational Displays for Taylor Ladder Logic .....	103
Segment Display .....	103
Segment Display Fields and Buttons .....	105
Faceplates for Program Elements.....	106
Register Display .....	108
Register Faceplate .....	109
Counter Display .....	110
Counter Faceplate.....	111
Timer Display .....	112
Timer Faceplate .....	114
I/O Point Display .....	115
I/O Point Faceplate .....	115
File Display .....	117
Sequencer Display .....	118
TLL Device Logged Changes .....	119
TLL Messages .....	119

## **Section 6 - Status Displays**

Status Displays Overview.....	121
System Status Display .....	122
Subsystem Types.....	123
Address .....	123
Subsystem Name .....	123
Subsystem Status.....	123
State Definitions.....	124
Subsystem Status Displays .....	124
AC460 Subsystem Status Display.....	125
AC410 Subsystem Status Display.....	130
Controller Node Subsystem Status Display .....	131
Controller Subsystem Status Display CCR Requests .....	131
Controller Subsystem Status Display Requests .....	132
Controller Subsystem Status Display Description .....	133
Turbo/Console Node Subsystem Status Display.....	137

Diagnostic Message Display .....	139
Device Identification.....	140
Archive Messages .....	140
Diagnostic Messages/Archive Display Description .....	140
Diagnostic Messages Display Targets .....	141
Printing Diagnostic Messages.....	143
System Performance Display .....	144
Reset ALL.....	145
Reset CPU .....	145
Reset CCF .....	145
Reset Memory Utilization.....	145

## **Section 7 - I/O Displays**

I/O Displays Overview.....	147
S800 I/O Displays .....	148
S800 LAN Display.....	148
AF100 INTERFACE - X Status Area.....	149
AF100 INTERFACE - X Menu.....	150
S800 Station Detail .....	151
S800 Station Display .....	152
FCI - X Status Area .....	154
FCI - X Menu.....	154
S800 Device and FCI Detail .....	155
S800 Device Display .....	156
S800 Warmstart .....	159
S100 I/O Displays .....	162
S100 LAN Display.....	162
S100 Device Status Display.....	165
Smoothstart Start-up Sequence .....	166
S100 Warmstart .....	167
PROFIBUS Displays.....	170
PROFIBUS LAN Display.....	170
PROFIBUS LAN Header Area.....	171

PROFIBUS LAN Status Area .....	171
PROFIBUS LAN Device Detail .....	172
PROFIBUS Device Display .....	174
PROFIBUS Device Header Area .....	175
PROFIBUS Device Status Area.....	176
Device Detail .....	177
Module/Channel Display .....	178
PROFIBUS Module/Channel Header Area .....	179
PROFIBUS Module/Channel Status Area .....	180
Startup .....	180
TRIO Displays.....	181
Accessing the Remote I/O Displays for a SC Controller .....	181
Accessing the Remote I/O Displays for an AC Controller .....	181
TRIO LAN Display.....	182
TRIO LAN Header Area.....	183
TRIO LAN Status Area.....	184
TRIO Block Display .....	188
TRIO Block Header Area.....	190
TRIO Block Status Area .....	190
TRIO Block I/O Area.....	191
Direct I/O Displays.....	193
Direct I/O Block Header Area.....	194
Direct I/O Area.....	194

## Section 8 - Operation Examples

Typical Operator Activities .....	197
Selecting and Viewing a Control Loop .....	198
Single Loop Operation in Automatic Mode .....	200
Single Loop Automatic Operation with Local Setpoint.....	200
Single Loop Automatic Operation with Remote Setpoint .....	201
Single Loop Automatic Operation with Ratio and Bias.....	202
Single Loop Automatic Operation with Feedforward.....	203
Single Loop Auto/Manual Transfer.....	205

Transfer from Manual to Automatic .....	205
Transfer from Automatic to Manual .....	207
Single Loop Operation in Manual Mode .....	208
Cascade Operation .....	209
Cascade Operation in Manual .....	209
Cascade Operation in Auto .....	210
Viewing Alarms on MOD Loop Displays .....	212
CCF Control Loop Faceplate with Alarms .....	212
Setpoint/Process Deviation Alarm Display .....	213
Viewing Abnormal State on MOD Loop Displays .....	214

## **Appendix A - Runtime Templets**

General .....	215
Loop Definition Templet .....	215
Device Loop Templet .....	223
PID Controller FCM .....	233
Adaptive Gain .....	247
Adaptive Reset .....	251
Auto/Manual Controller FCM .....	252
Input FCMs .....	253
Common Fields .....	253
Analog Input FCM .....	253
Digital Input FCM .....	254
Pulsed Input FCM .....	254
Pulsed Input Time Derivative FCM .....	254
Output FCMs .....	255
Common Fields .....	255
Analog Output FCM .....	256
Digital Output FCM .....	257
Pulse Duration Output FCM .....	257
Pulse Train Output FCM .....	257
Common Calculator FCM Fields .....	258
Math Related Calculator FCMs .....	259

Sum of 4 Inputs FCM.....	259
Polynomial FCM.....	259
Average FCM.....	259
Subtraction FCM.....	260
Division FCM.....	260
Natural Logarithm FCM .....	260
Exponentiation FCM .....	260
Multiplication FCM .....	261
Absolute Value FCM .....	261
Scale Compensation Calculator FCMs.....	261
Modified Square Root FCM.....	261
Linearization FCM .....	262
Normalize FCM .....	263
Inverse Normalize FCM.....	263
Scale Input FCM .....	264
Flow Calculation FCM.....	264
Temperature Compensation FCM.....	265
Ratio/Bias FCM .....	266
Time Related Calculator FCMs.....	266
Time Derivative FCM .....	267
Time Integration FCM .....	267
First Order Filter FCM.....	267
Totalizer FCM.....	268
Lead/Lag Filter FCM .....	268
Dead Time FCM .....	269
Dead Time Compensation FCM .....	269
Timer FCM.....	270
Counter FCM .....	270
Delay Timer FCM.....	271
Logic Calculator FCMs .....	271
Logical AND.....	271
Logical OR.....	272

Logical NOT .....	273
Exclusive OR .....	273
Set/Reset Flip-Flop .....	274
Real Compare FCM .....	274
Miscellaneous Calculator FCMs .....	275
Selector FCM .....	275
Put Generic Value FCM .....	276
Get Generic Value FCM .....	277
Select Next .....	278
Redundant Signal Selector .....	279
Input Limiter FCM .....	279
User Calculations FCM .....	280
Data Entry FCM .....	280
String FCM .....	280
User Math Block FCM .....	281
MOD30_MOD300_MAP .....	282
Extended Processing FCM .....	282
Common Fields .....	282
Continuous Moving Average FCM .....	284
Standard Deviation FCM .....	284
Non-Rate Periodic Total FCM .....	284
Periodic Average FCM .....	284
Periodic Maximum FCM .....	285
Periodic Minimum FCM .....	285
Periodic Rate Total FCM .....	285
S800 I/O Runtime Templets .....	285
AI810, AI890 .....	286
AI820 .....	288
AI830 .....	291
AI835 .....	294
AO810, AO890 .....	297
AO820 .....	300

DI810 .....	303
DI814 .....	306
DI820 .....	307
DI821 .....	309
DO810 .....	311
DO814 .....	315
DO820 .....	317
Special Applications Runtime Templets .....	320
SP_ALGO1_FCM.....	320
SP_ALGO2_FCM.....	320
SP_ALGO3_FCM.....	320
SP_ALGO4_FCM.....	320
SP_ALGO5_FCM.....	320
BRKPTS .....	321
NAME OF SET .....	321
X VALUE .....	321
Y VALUE .....	321

## INDEX





---

# About This Book

## General

This book describes how to operate 800xA for MOD 300 software. MOD 300 integration software provides standard display formats on the 800xA System that are similar in features and functions to the Multibus based MOD 300 and Unix based Advant OCS systems. Use this section as a guide to the information presented in this book. Related documentation references are also provided.

## Use of Warning, Caution, Information, and Tip

This publication includes **Warning**, **Caution**, and **Information**, where appropriate, to point out safety related or other important information. It also includes **Tip** to point out useful hints to the reader. The corresponding symbols should be interpreted as follows:



Electrical warning indicates the presence of a hazard which could result in *electrical shock*.



Warning indicates the presence of a hazard which could result in *personal injury*.



Caution indicates important information or warning related to the concept discussed in the text. It might indicate the presence of a hazard which could result in *corruption of software or damage to equipment/property*.



Information alerts the reader to pertinent facts and conditions.



Tip indicates advice on, for example, how to design your project or how to use a certain function.

---

Although **Warning** hazards are related to personal injury, and **Caution** hazards are associated with equipment or property damage, it should be understood that operation of damaged equipment could, under certain operational conditions, result in degraded process performance leading to personal injury or death. Therefore, comply fully with all **Warning** and **Caution** notices.

## Document Conventions

The following conventions are used for the presentation of material:

- The words in names of screen elements (for example, the title in the title bar of a window, the label for a field of a dialog box) are initially capitalized.
- Capital letters are used for the name of a keyboard key if it is labeled on the keyboard. For example, press the ENTER key.
- Lowercase letters are used for the name of a keyboard key that is not labeled on the keyboard. For example, the space bar, comma key, and so on.
- Press CTRL+C indicates that you must hold down the CTRL key while pressing the C key (to copy a selected object in this case).
- The names of push and toggle buttons are boldfaced. For example, click **OK**.
- The names of menus and menu items are boldfaced. For example, the **File** menu.
  - The following convention is used for menu operations: **MenuName > MenuItem > CascadedMenuItem**. For example: choose **File > New > Type**.
  - The **Start** menu name always refers to the **Start** menu on the Windows Task Bar.
- System prompts/messages are shown in the Courier font, and user responses/input are in the boldfaced Courier font. For example, if you enter a value out of range, the following message is displayed:

Entered value is not valid. The value must be 0 to 30.

You may be told to enter the string TIC132 in a field. The string is shown as follows in the procedure:

---

## TIC132

Variables are shown using lowercase italic letters:

*sequence name*

## Terminology

You should be familiar with the following list of terms used in this instruction.

<b>Term</b>	<b>Description</b>
AC410, AC460	Advant Controller 410 and 460
AdvaBuild	Configuration package for Advant OCS running on an Engineering Station (HP-UX or Windows)
Advant OCS	Advant Open Control System or MOD control system
Aspect	An aspect is a description of some properties of an object. The described properties could be mechanical layout, how the object is controlled, a live video image, name of the object and so forth.
Aspect Object	An aspect object works like an information container for it's aspects.
CCF	Configurable Control Functions.
Client	Client is a part of a software that subscribes data from a server.
Connectivity Server	A server that provides access to controllers and other sources for real-time data, historical data, and alarm and event data. A Connectivity Server runs services related to OPC/DA, OPC/AE, OPC/HDA.
DCN	Distributed Communication Network. Connects MOD control system nodes.

---

Term	Description
Faceplate	A faceplate is an aspect that provides a graphical representation of a certain aspect object, with presentation of certain properties related to the object, and mechanism for operator interaction such as on/off, increase/decrease, and so forth.
FCM	Function Class Module
Node	A point of interconnection to a network typically having a unique node address. Every physical computer box with its own host name is considered a node.
Node level MOD connections	CNTRLLER, CONSOLE, GENERICD, AC410, AC460MOD, BUC, DCN_DCN, ADVANT_D2D
Object	Any configurable entity whose data is associated with some real object such as a valve or transmitter, an order or a virtual object like a service.
Primary Aspect	The view assigned to an object that can be opened by double clicking the object (i.e., double clicking and event on an event page opens a faceplate).
TCL	Taylor Control Language (MOD sequence control)
TLL	Taylor Ladder Logic
TRIO	Taylor Remote I/O
User	The person using the system. This could be an engineer building the system or an operator controlling a process via the system.
Windows	Microsoft Windows Operating System (as used in this book).

---

## Related Documentation

This instruction provides information only for the operation of 800xA for MOD 300. The following lists additional related documentation.

*Table 1. Related Documentation*

<b>Category</b>	<b>Title</b>	<b>Description</b>
Hardware	<i>Installation Instructions for Real-time Accelerator Board</i>	Details regarding installation of RTA Board.
	applicable computer user's guide	Details regarding installation of computer.
Installation	<i>Industrial IT 800xA System Installation</i>	Describes the installation of the base system software.
Configuration	<i>Industrial IT 800xA System Configuration</i>	Describes the configuration of the base system after software has been installed.
	<i>800xA for MOD 300 Configuration</i>	Describes the configuration of the MOD 300 functions.
Operation	<i>Industrial IT 800xA System Operation</i>	Describes the user interface of the base product.
	<i>800xA for MOD 300 Operation</i>	<u>This Document</u> . Describes the user interface of the MOD 300 displays.

*Table 1. Related Documentation*

<b>Category</b>	<b>Title</b>	<b>Description</b>
MOD 300 Reference	<i>Configurable Control Functions (CCF) User's Guide</i>	Describes the Configurable Control Functions software and the techniques needed to implement them and is designed to help you build your control schemes in the Advant OCS.
	<i>Taylor Control Language (TCL) User's Guide (AdvaBuild v2.7 to v3.3)</i>	Provides format and syntax rules, and other reference information for TCL.
	<i>Taylor Ladder Logic (TLL) Users Guide (Supports AdvaBuild Versions 2.7 to 3.3)</i>	Provides format and syntax rules, and other reference information for TLL.
	<i>Data Base Tables</i>	Contains information about the data base of the MOD 300 System to aid in configuration, TCL programing, display building and similar activities.
	<i>Diagnostic Error Messages</i>	Listing of Error Messages.

## On-line Documentation

All documentation is supplied in Adobe® Acrobat® reader (.pdf) format. Acrobat Reader is a freeware program, which can be downloaded from the Internet. You can get it from: <http://www.adobe.com>. Windows must be set up to start the Acrobat Reader for .pdf files.

On-line documentation is available on the MOD installation CD and is accessible from the Documentation Structure of the 800xA system if the documentation option is installed. The installed documentation for 800xA for MOD 300 software includes:

- 800xA for MOD 300 Configuration
- 800xA for MOD 300 Operation

## On-line Help

Configuration and operation help is included in an HTML compiled format. Help for Operation and for the MOD Importer is context sensitive.





---

# Section 1 Introduction

## Document Overview

This book describes how to use 800xA for MOD 300 software for MOD 300 system operation. The 800xA System provides a common operation and configuration view for MOD 300 and other field devices and software options. This book has the following objectives:

- How to use 800xA for MOD 300 displays and faceplates
- Operating examples
- Provide additional reference information as necessary

## Intended User

As a prerequisite you should already be familiar with the MOD 300 control system you will be working with. In addition, you must have operator privileges on the computer where the software runs. This book is not intended to be the sole source of instruction for the software. It is highly recommended that those people who will be involved in system operation attend the applicable training courses offered by ABB.

The following are some quick guidelines to help you find what you are looking for in this book.

## Where to Start

Refer to [Section 2, Getting Started](#) for instructions on starting 800xA for MOD 300 software and some basic operations within the 800xA operations framework.

## Operation

Refer to the following operation chapters for instructions and reference information related to runtime operation. Specifically, these chapters provide a detailed description of each display related to runtime operation.

- [Section 3, CCF Displays](#)
- [Section 4, TCL Displays](#)
- [Section 5, TLL Displays](#)

## Diagnostic Displays

[Section 6, Status Displays](#) contains information associated with using the diagnostic and status displays.

## I/O Displays

[Section 7, I/O Displays](#) contains information associated with using the I/O displays.

## Operation Examples

[Section 8, Operation Examples](#) contains information intended to help the user better understand how to use the displays.

## On-line Help

Comprehensive on-line help is available for the software. Access to the complete help files is available via the **Help** menu in the main menu bar. Context-sensitive help (**F1**) is available for some windows.

## Product Overview

800xA for MOD 300 is an integration product to the Industrial IT 800xA System for the MOD 300 control network (DCN). The Connectivity Server, with a Real Time Accelerator Board (RTAB) connects to the DCN and gathers data for display as different MOD Aspect Objects on Client Workstations.

The system offers features and functions designed to facilitate and optimize any process or enterprise. Information is presented in an intuitive form using standard display aspects similar in features and functions to the Unix and Multibus based Advant OCS system. Customized user accounts allow operators to deal with operating conditions in a productive manner.

The interface used allows multiple aspects of objects to be viewed in a format that is familiar to users. Operation of the control system works within this interface to give you the ability to easily explore Aspect Objects, control loops, acknowledge abnormal conditions and perform many other activities necessary to monitor and control a process.

The MOD 300 display aspects are:

- System and Status
  - MOD 300 System Status
  - MOD 300 AC410 CPS Status
  - MOD 300 AC460 CPS Status
  - MOD 300 Controller Subsystem
  - MOD 300 Multibus
  - MOD 300 Message
  - MOD 300 System Performance
  - MOD 300 Alarm Summary
- CCF
  - MOD 300 Loop Detail
  - MOD 300 Loop FCM
  - MOD 300 Environment Area Graphic
  - MOD 300 Environment Area Status
  - MOD 300 Environment Group Graphic
  - MOD 300 Environment Group Status
  - MOD 300 Environment Group Trend

- TCL
  - MOD 300 TCL Unit Overview
  - MOD 300 TCL Unit Detail
  - MOD 300 TCL Recipe Detail
  - MOD 300 TCL Sequence Debug
  - MOD 300 TCL Sequence Detail
  - MOD 300 TCL SFC
  - MOD 300 TCL Unit Array Plot
  
- TLL
  - MOD 300 TLL Counter
  - MOD 300 TLL Counter Faceplate
  - MOD 300 TLL File
  - MOD 300 TLL I/O Point
  - MOD 300 TLL I/O Point Faceplate
  - MOD 300 TLL Register
  - MOD 300 TLL Register Faceplate
  - MOD 300 TLL Segment
  - MOD 300 TLL Sequencer
  - MOD 300 TLL Timer
  - MOD 300 TLL Timer Faceplate

---

## Section 2 Getting Started

### General

Installation information for the base product is in *Industrial IT 800xA System Installation*. Installation of the displays described in this book is in *800xA for MOD 300 Installation*.

Configuration information for the base product is in *Industrial IT 800xA System Configuration*. Configuration of the displays described in this book is in *800xA for MOD 300 Configuration*.

### Start-up Procedure

Start the workplace as follows.

1. Turn on your computer and wait for the logon screen.
2. Enter your operator user name and password and select OK.

Your status changes to the User Rights as defined in the User Manager and you are now logged in to the computer.

3. Select **Start > Programs > ABB Industrial IT > Start Workplace**

The Workplace Login is launched. Start the Plant Explorer or Operator Workplace to access the MOD 300 displays.

### Product Verification

With the workplace started, select the **About Industrial IT** icon . The About screen identifies the version of the installed products and extensions.

## Log on as a Different User

The workplace uses Windows administration rights as the basis for its security system. Operators have specific rights necessary to start and use the workplace. The user rights associated with your logon name and password determines your security level.

To log into the client station as an operator:

1. Select **Start > Shut Down...** to get the Shut Down Windows display.
2. Select “Close all programs and log on as a different user?” if prompted.
3. Enter your operator User name and Password and select OK.

Your status changes to the User Rights as defined in the User Manager and you are now logged in to the client station.

The Log Over function enables a fast and temporary switch between users in a running workplace. The log over changes the permissions and user roles but keeps all open windows with their present contents. The permitted actions in the open windows are controlled by the permissions of the logged over user. The log over only affects the 800xA System permission. Windows security is still the same as the user logged in.

To log over into the 800xA System as a new user:

1. Right click on the user name in bottom line of your workplace window and select **Change User**. A Change-user authentication dialog box appears.



If MOD specific displays are open, a Close View window will appear and the listed MOD 300 displays must be closed before the user can be logged over.

2. Enter your User ID with domain and password. If accepted, the dialog box disappears and the new user can operate the workplace.
3. To return to the first user right click on the user name again and select **Revert User**. The **Revert User** operation requires authentication of the user to revert to.

## Operating Overview

The control packages (CCF, TCL, and TLL) provide automatic control functions that may or may not involve operator interaction, depending upon the requirements of each particular application. When operator interaction is required, the operator interface is supported by the displays described in the following chapters. These displays are based upon aspects of the object type templets described in *800xA for MOD 300 Configuration*. See the *Industrial IT 800xA System Operation* book for basic topics regarding runtime operation.

The following major functions are supported by the operator interface:

- Values of parameters from the loops are displayed in numerical and graphic form
- Operator can change certain parameter values such as setpoints, outputs, setpoint modes, output modes, and device commands from the console
- Display and acknowledgment of alarm conditions
- Operator can change (tune) some aspects of the configuration while the system is operating

## Accessing Displays

Use the aspect browser, display associations and links to access displays described in this book.



Loop Templet and FCM Templet displays are accessed from the Loop FCM display. However, these displays are case sensitive to the tag name. If a valid tag is SC5\_3\_PID1, entering Sc5\_3\_PID1 will not access the templet displays.

## Accessing Displays Using Browser

The aspect browser, in the form of a tree-type control, is similar to Windows Explorer and is used to access objects for display. Defined objects in this nested view are categorized within logical structures. For example, MOD 300 objects get imported into the Control Structure.

## Accessing Displays Using Graphic Displays

Graphic displays may be associated with an object with context navigation to trends, faceplates, and overview type displays. Overview graphics can provide a panoramic view of the process and can provide one-step access to critical displays. They can be made the software equivalent of the Page Selector Alarm Panel with the capability to monitor alarms for different areas and groups from one display or have target blocks that can initiate a TCL program sequence state or status transition.

## Accessing Displays Using Context Menus

Context menus, [Figure 1](#), provide display access based upon the object type or upon the selection in the current display (for example, access to the Lan Display or Station Display when working with S800 I/O). Use the right mouse button to get an applicable context menu. User roles affect the contents of the context menu. Also, these menus can be filtered to show only the MOD 300 specific aspects.

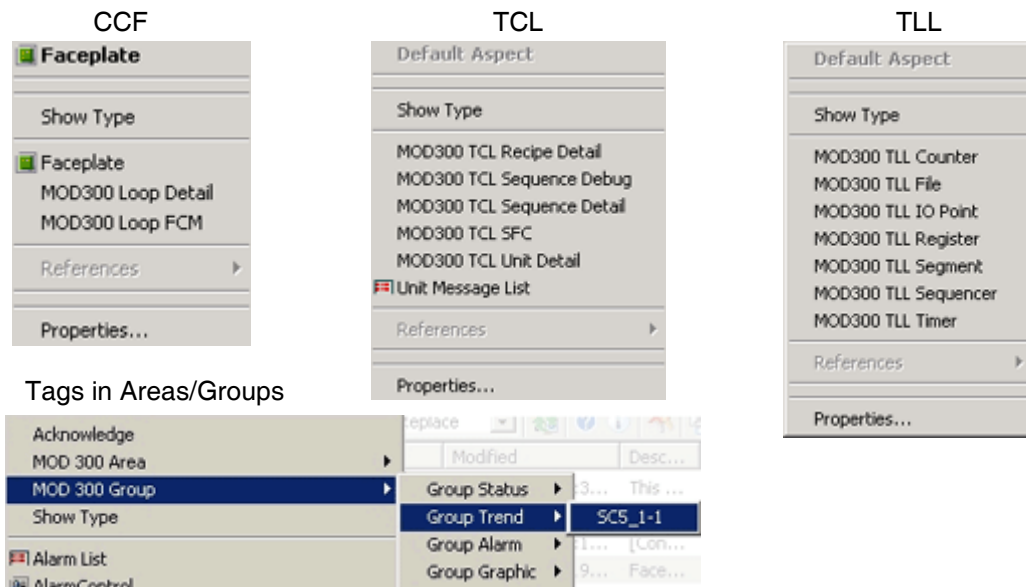


Figure 1. Context Menus (Typical) for MOD Tag Aspects



## MOD 300 Objects and Associated Aspects

Table 2 shows the relationships between MOD 300 objects and aspects.

Table 2. MOD 300 Objects and Associated Aspects

MOD 300 Objects	Associated Aspects
MOD_DB	MOD 300 System Status, Alarm and Event Lists (CCF Alarm, Diagnostic Message, TCL Message)
MOD_AREA	MOD 300 System Status
MOD_AC460	MOD300 AC460 PS Status, MOD 300 System Status
MOD_AC460MOD	MOD300 AC460 PS Status, MOD300 Message, MOD300 System Performance, MOD 300 System Status
MOD_AC410	MOD300 AC410 PS Status, MOD300 Message, MOD300 System Performance, MOD 300 System Status
MOD_CCF MOD_CTRL_BLOCK	MOD 300 System Status
MOD_CCF_CONTIN_LOOP	MOD300 Alarm Summary, MOD300 Loop Detail, MOD300 Loop FCM, Faceplate, Alarm List, Measure_DVAlue
MOD_CCF_CNTRL_LOOP	MOD300 Alarm Summary, MOD300 Loop Detail, MOD300 Loop FCM, Faceplate, Alarm List, Measure_DVAlue Output_DValue
MOD_CCF_PID_LOOP	MOD300 Alarm Summary, MOD300 Loop Detail, MOD300 Loop FCM, Faceplate, Alarm List, Measure_DVAlue, Output_DValue, Setpoint_DValue

Table 2. MOD 300 Objects and Associated Aspects

MOD 300 Objects	Associated Aspects
MOD_CCF_DEV_LOOP	MOD300 Alarm Summary, MOD300 Loop Detail, MOD300 Loop FCM, Faceplate, Alarm List, Command_DVAlue
MOD_UNIT	MOD300 TCL, MOD300 Recipe Detail, MOD300 Sequence Debug, MOD300 Sequence Detail, MOD300 SFC, MOD300 Unit Detail
MOD_SEQUENCE	MOD300 Sequence Debug, MOD300 Sequence Detail, MOD300 SFC
MOD_LL_DEV	MOD300 TLL, MOD300 Counter, MOD300 File, MOD300 IOPoint, MOD300 Register, MOD300 Segment, MOD300 Sequencer, MOD300 Timer
MOD_LL_CNTR_GRP, MOD_LLIO_GRP, MOD_LL_REG_GRP, MOD_LL_TIMER_GRP	MOD 300 System Status
MOD_LL_CNTR	MOD300 TLL Counter Faceplate
MOD_LL_I_O	MOD300 TLL IO Point Faceplate
MOD_LL_REG	MOD300 TLL Register Faceplate
MOD_LL_TIMER	MOD300 TLL Timer Faceplate
MOD_D2F, MOD_ADVANT_STATION	MOD300 Message, MOD300 System Performance, MOD 300 System Status
MOD_CONT_SS	MOD300 Controller Subsystem, MOD 300 System Status
MOD_CONTRLLER, MOD_BUC	MOD300 Controller Subsystem, MOD300 Message, MOD300 System Performance, MOD 300 System Status

Table 2. MOD 300 Objects and Associated Aspects

MOD 300 Objects	Associated Aspects
MOD_BUM	MOD300 Message, MOD 300 System Status
MOD_CONSOLE, MOD_ADVANT_D2D, MOD_DCN_DCN, MOD_GENERICD	MOD300 Multibus, MOD300 Message, MOD300 System Performance, MOD 300 System Status
MOD_ENV_AREA	MOD300 Area Status, MOD300 Area Graphic, MOD300 Area Alarm
MOD_ENV_GROUP	MOD300 Group Status, MOD300 Group Graphic, MOD300 Group Alarm, MOD300 Group Trend

## Availability of Displays as Indicated by Server Status

The top right corner of each MOD display has one or two (redundancy) indicators that show the server status between the client/server and the server/controller. These indicators show the status conditions as described in [Table 3](#). An overview display with multiple controllers will not go yellow unless all nodes are down.

Table 3. Server Status Indicators

Condition	Indication	Description
OK	Green	Connection between client/server is good.
Failed	Red	A connection was made but is not working.
Not Connected	White	No connection made between client/server.
Trying	Blue	Attempting a connection between client/server.
Node Down	Yellow	The server/controller connection has failed. The controller node may be down.

## Printing Displays

To print the active window of a display, use the Print Screen key to capture the screen, paste the image into an application such as Microsoft Word and then send it to the printer. For best resolution, maximize detail displays to full screen before printing.

By default, the MOD Environment Group Trend, Group Status, Group Graphic and Area Status print in the landscape mode for best orientation.

## Alarm/Event Handling

MOD 300 has multiple alarm conditions per object (Measure, Setpoint, Output, Deviation and so forth). To handle these properties, a MOD Alarm/Event List is provided as an aspect. Alarm/Events are handled as follows.

Alarm/Event types with user action (acknowledgement):

- CCF (Measure, Setpoint, Output, Deviation, Device)
- TCL (Message, Reply, Error, Unit Alarm)
- Diagnostics
- User Ack (outgoing)
- TCL Reply(outgoing)
- Global Ack broadcast (incoming)
- Global Ack broadcast (outgoing)

Event types with no user action required (for history/loggers only):

- Parameter Change (incoming) - NOTE: Parameter change messages for template and faceplate changes always come through as Administrator.
- User Log On/Off (incoming)
- User Ack (incoming)

---

## Section 3 CCF Displays

### Operational Displays for Configurable Control Functions

The format and functionality of each runtime display that supports the Configurable Control Functions (CCF) is described here along with procedures for basic operations that are executed from the displays. In addition, tuning and engineering procedures are described. The runtime displays that support CCF are:

- [Loop Detail](#)
- [Loop Faceplate](#)
- [Loop FCM](#)
- [Loop Templets](#) ([Appendix A, Runtime Templets](#))
- [Area Displays](#) (Alarm, Status, and Graphic)
- [Group Displays](#) (Trend, Alarm, Status, and Graphic)

These basic operational displays are supported by runtime versions of the Loop Definition and FCM Templets through the Loop FCM display. Continuous loops perform indication, and calculation functions. Control loops provide output control and PID loops provide output control with a setpoint. Device loops control discrete devices such as fans and motors. The templets for these loops are described in the *Configurable Control Functions (CCF) User's Guide*.

### Loop Detail Displays

The Loop Detail Display provides the means to manipulate tunable parameters. The information available on this display varies according to the loop type: Control, Continuous, PID, and Device.



Generally, users with Operator authority have read-only access to the Loop Detail Display. Control/tuning functions are restricted to System Engineers. The authority assignment for functions associated with this display can be changed as described in the Objects and Access Rights section of *Industrial IT 800xA System Configuration*.

## Control and PID Loops

The Loop Detail Display for a PID loop, [Figure 2](#), shows the measured value, setpoint value, and output value trends for a loop. A reduced or standard size faceplate is used to modify the setpoint and output values. The trend graph shows up to 600 points (traces start when the window is displayed). The data is based on CCF trend data.

The trend **Refresh** rate on the Loop Detail page, allows you to change the rate at which a trend graph is updated. Select **Refresh** to get the **Change Refresh Rate** dialog where you can change this rate to any integer number from 1 to 120 seconds, the default value is 1. After changing the Refresh value, the plot restarts and the trend is refreshed at the new rate.

The Loop Detail Display also provides the means to set the following loop parameters:

- Loop setup values, including Phase, Scan Rate, Loop Mode, and Trend Rate
- Tuning parameters, including Gain, Reset, and Preact
- Alarm configuration conditions
  - Alarm Post
  - Alarm Check
  - Engineering Deadband
  - Output Deadband
  - Deviation Deadband
  - Enabled/disabled condition, limit value where applicable, and priority for each alarm parameter (input, setpoint, deviation and output)

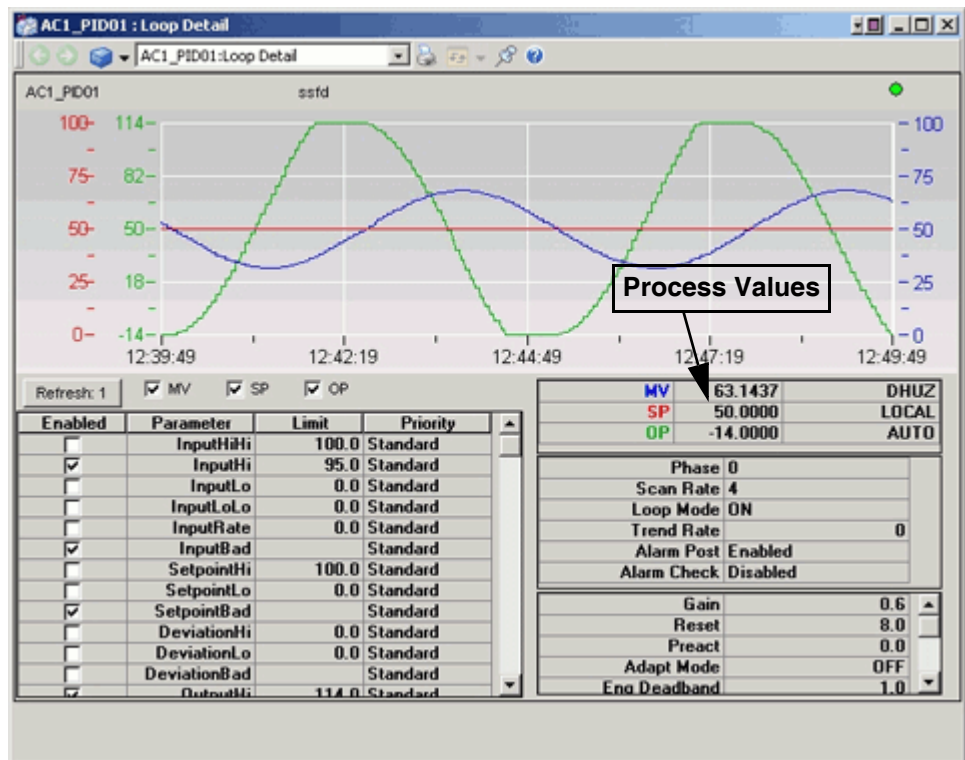


Figure 2. Loop Detail Display for a PID Loop

If you click on any field in the process values box, a Faceplate, [Figure 3](#), is displayed. This faceplate enables you to change the setpoint and output values displayed for the loop in the Loop Detail Display as well as other values and modes.

Reduced size faceplates, [Figure 3](#), provide the basic operator control actions without the process bar graphs of the standard size faceplate, [Figure 4](#). The reduced size faceplates for PID loops show measure, setpoint, output and other parameters as determined by configuration such as bias, ratio, computer mode, feedforward and feedback. Similarly, the Auto/Manual Ration/Bias (AMRB) loop does not show the setpoint, [Figure 5](#).

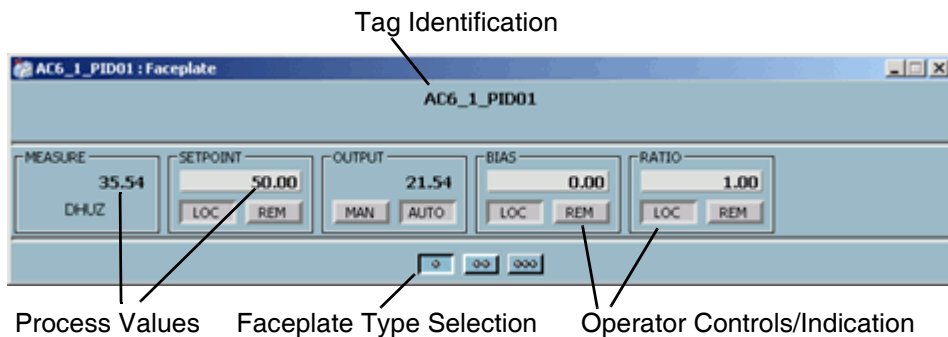


Figure 3. MOD CCF PID Loop Reduced Faceplate

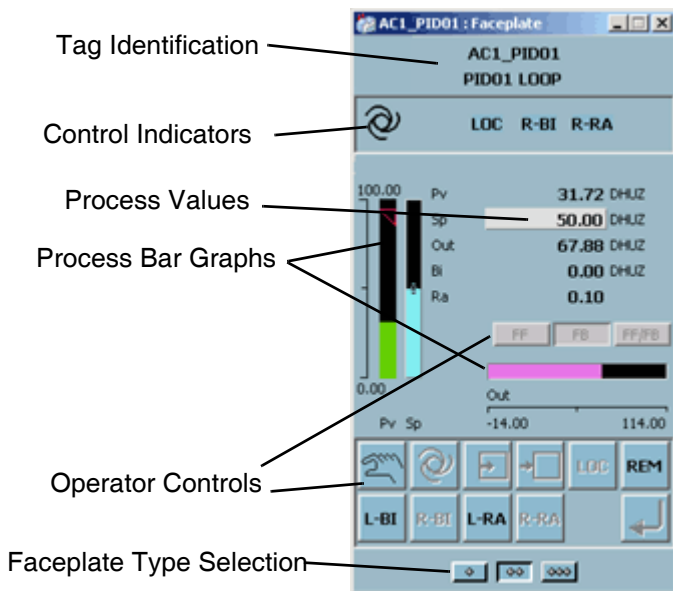


Figure 4. MOD CCF PID Loop Standard Faceplate



Fields that do not apply to the type of loop being displayed are left blank and the labels for these fields are dimmed. For Auto/Manual Controller loops, Figure 5, the following fields are blank: setpoint; setpoint and deviation alarms; deviation deadband; and tuning values. See Chapter 8, Operation Examples, for explanations on the use of the various loop faceplate types.



Figure 5. A/M Ratio/Bias Controller Faceplates

## Device Loops

Figure 6 is an example of the information that is included on a Loop Detail Display for device loops. Device loops detect the device state. There is no trend or alarm limit/priority information on Loop Detail Displays for device loops. Select the command, mode, and state process values box to access the device loop faceplate. This display also provides the means to set the following device loop parameters.

- Loop setup values, including: Phase, Scan Rate, Loop Mode, Simulation Mode, Override Mode, and Lock State. Field State is a read only field and cannot be changed.
- Alarm configuration conditions, including Alarm Post, and Alarm Check.

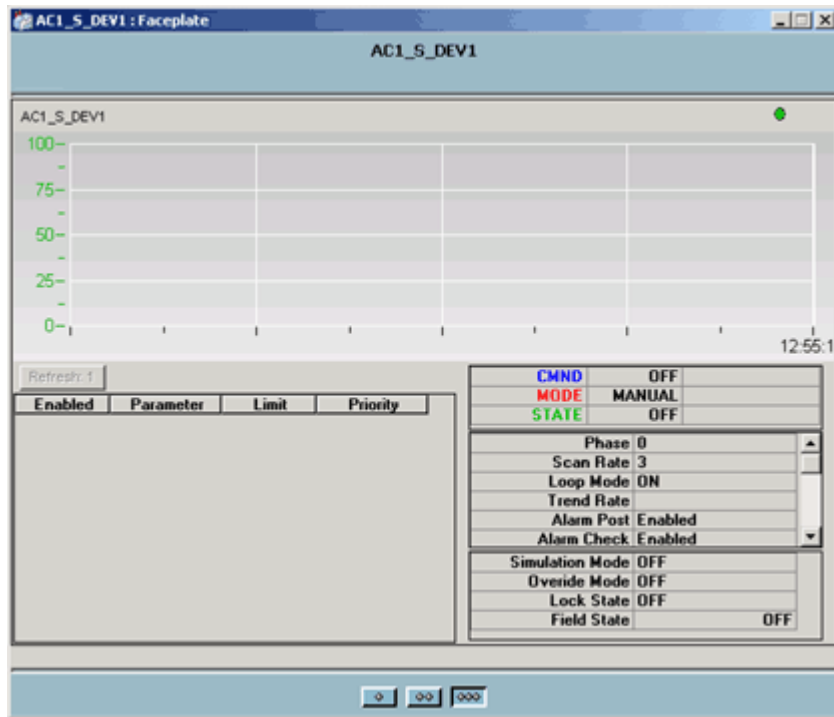


Figure 6. Loop Detail Extended Faceplate for a Device Loop

The device loop standard faceplate, [Figure 7](#), is used to modify command, mode, and state. Only device descriptor state values of 0 to 15 are allowed.

If the Enable Access Restriction configuration is set to Yes, then device commands are not allowed while in auto (shaded buttons). If the Manual Enable configuration is set to Disallowed, then the Manual button is shaded to prevent changes back to manual (may already be in manual before the configuration change). Allow the Manual Enable configuration change to process before making mode changes.

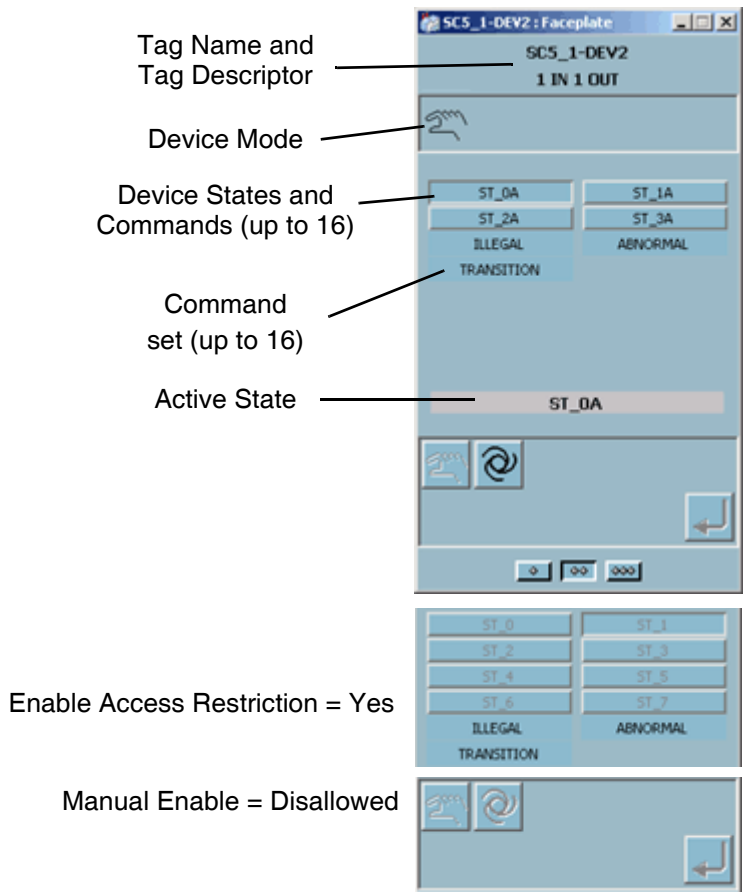


Figure 7. MOD CCF Device Loop Standard Faceplate

The device loop reduced faceplate, [Figure 8](#), is also used to modify command, mode, and state.



Figure 8. MOD CCF Device Loop Reduced Faceplate

## Continuous Loops

Indicator loops measure the values of process variables. The types of information that are included on a Loop Detail Display or a faceplate for indicator loops are: tag, descriptor, measured variable, engineering units and alarm indications.

[Figure 9](#) is an example of the types of information that are included on a faceplate for a calculator type continuous loop. Calculator loops perform calculations on their inputs. You can use the results of the calculations in other loops.

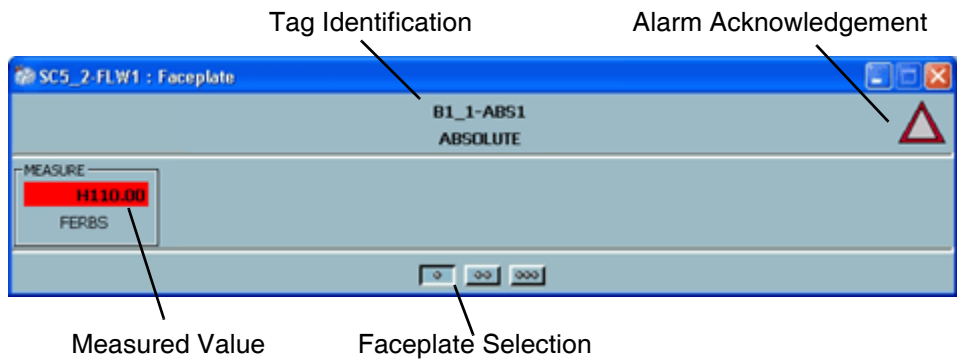


Figure 9. MOD CCF Reduced Faceplate for Continuous Loop

## Loop Faceplates

There are three different faceplate types that support different levels of information, [Figure 5](#). The reduced size offers basic operator control, the standard size includes process bar graphs and additional operator controls, and the extended size offers tuning parameters in the form of the loop detail display. These reduced size and standard size faceplates are described below. The extended size is discussed in [Loop Detail Displays](#) on page 37.

### Tag Identification and Indicators

Contains the tag name, tag descriptor and abnormal state indication, [Figure 10](#). A tag name can be up to 12 characters and is defined during tag configuration. This identifies the tag for monitoring and control of the process point. A tag descriptor can be up to 24 characters and normally explains the purpose of the tag. The descriptor defined during configuration and then imported into the 800xA system, after it is imported it is no longer updated.

An abnormal condition for a continuous loop is checked when the [ABNORMAL STATE Field](#) on page 221 is set to yes.

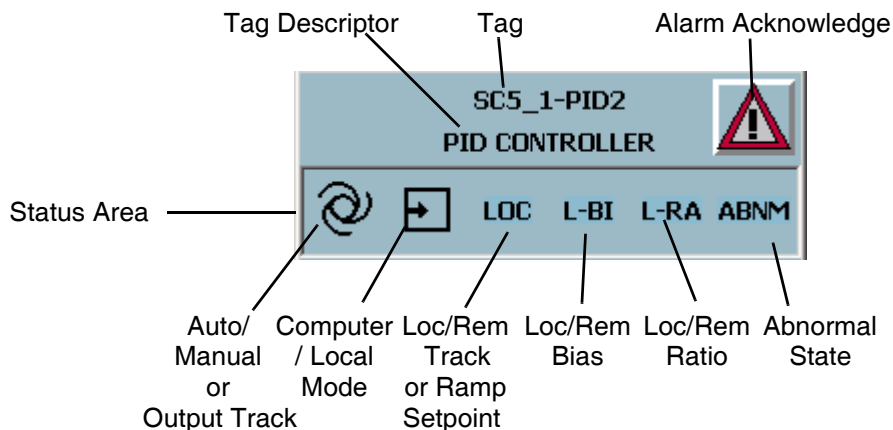


Figure 10. Faceplate Name Area Features

### Process Bar Graphs and Values

The area shown in [Figure 11](#) contains graphical and numerical representations of the tag being monitored. The bar graph area includes scaled indication bars for measured value (Pv), setpoint (Sp) and output (Out); slider controls to change setpoint and output; and alarm markers. The number area includes measured value (Pv), setpoint (Sp), control output (Out), setpoint high and low limits and ratio (Ra) and bias (Bi) values. You can use the number area to enter a value for operator controllable values (configured limits are enforced). Alarms are indicated using colored backgrounds for priority, flashing for unacknowledged, and the characters H (high), L (low), D (Deviation), R (Rate) and ? (Bad Data Quality) in front of the display value. Infinite values will show ?Inf (variations shown on other displays such as ?1.#IO for status blocks or 1.\$e+000 for Loop FCM). The background color corresponds to the priority color of the highest priority alarm currently active.

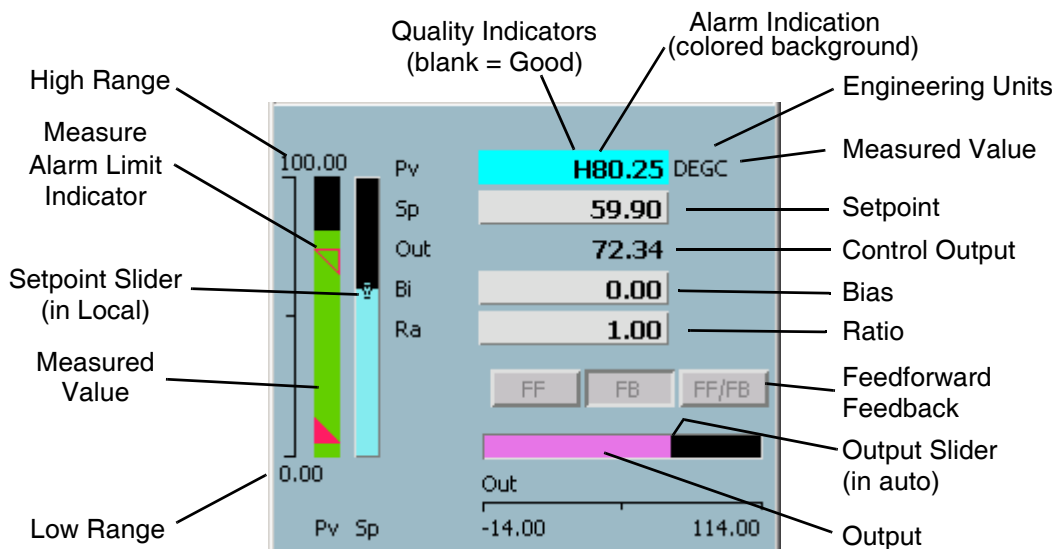


Figure 11. Faceplate Process Bar Graph and Values Features (CCF)

## Operator Controls

The Operator area, [Figure 12](#), provides controls that manipulate the tag being displayed. Some of the actions available in the operator area include: change auto or manual mode, change computer mode, and change set point mode (a shaded button does not allow change).

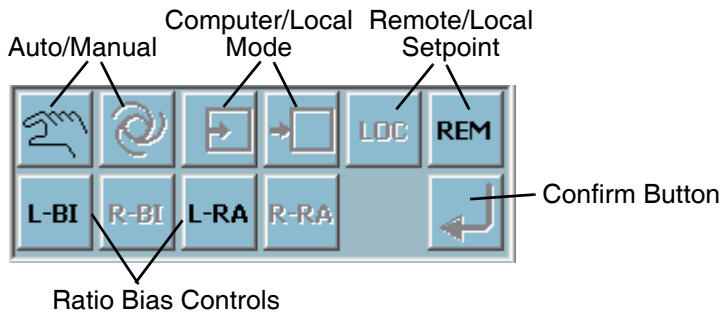
If the Confirm button is enabled, all commands in the operator area must be confirmed by clicking the confirm button when it is active. If Confirm is not clicked in a configured amount of time, the value will not change and the button will return to its normal color.

If the hide Confirm button option is enabled, the behavior of the faceplates change allowing a value to be entered immediately.

The signal indicators and set of state buttons and their definitions are given in [Table 4](#).



If an operator control is shaded, switching to Auto/Manual or Local/Remote is not allowed.



*Figure 12. Faceplate Operator Control Features*

Table 4. Buttons and Indicators







Icon	Name	Description
	<b>Auto</b>	Switch to Auto from Manual. In auto, the setpoint and ratio can be adjusted. If shaded, switching to Auto is not allowed.
	<b>Manual</b>	Switch to Manual from Auto. In manual, the output can be adjusted. If shaded, switching to Manual is not allowed.
	<b>Computer Mode</b>	Request a switch to Computer mode. Restricted parameters for PID loops are SETPOINT, SPT_MODE, RESULT and OUT_MODE. Restricted parameters for Controller loops are RESULT and OUT_MODE.  In <b>Computer Mode</b> , you have two options; LOCAL, where the operator is able to access the restricted parameters but the computer cannot, and COMPUTER, where the computer program is allowed access to the restricted parameters but the operator is not. When the COMPUTER option is selected, the operator can still change control to LOCAL.
	<b>Local Mode</b>	In Local Mode, the operator is able to access the restricted parameters but the computer cannot (see Computer Mode).
	<b>Remote</b>	Switch to remote setpoint. Remote setpoint value is the result of another FCM. If shaded, switching to Remote is not allowed.
	<b>Local</b>	Switch to local setpoint. Local setpoint value comes from within the PID FCM. The operator can change this value. If shaded, switching to Local is not allowed.



Table 4. Buttons and Indicators (Continued)















Icon	Name	Description
	<b>Local Bias</b>	Switch to Local Bias operation. Bias value can be changed locally while in this mode.
	<b>Remote Bias</b>	Switch to Remote Bias operation.
	<b>Local Ratio</b>	Switch to Local Ratio operation. Ratio value can be changed locally while in this mode.
	<b>Remote Ratio</b>	Switch to Remote Ratio operation.
	<b>FeedBack</b>	Feedback is used with the standard integral in the PID algorithm to provide an anti reset-windup capability. If the feedback input is not used, the limited output value is used.
	<b>FeedForward</b>	This control loop has a feed forward input that is added or multiplied to the denormalized AUTO output of the PID algorithm.
	<b>FeedForward-FeedBack</b>	Both FeedForward and FeedBack are used.
	<b>Control Output Tracking</b>	Tag is in control output tracking mode. If the station goes into output tracking mode, the output percentage adjusts to changes in its reference signal. Control output changes cannot be initiated from the workstation when the tag is in this mode.
	<b>Setpoint Tracking</b>	Tag is in setpoint tracking mode (LOC and REM will be shaded). The set point tracks either the process variable or a selected variable. The set point cannot be adjusted from the console in this mode.

Table 4. Buttons and Indicators (Continued)

Icon	Name	Description
	<b>Ramp</b>	Setpoint is being ramped by a TCL source.
	<b>Abnormal State</b>	An abnormal condition for a CCF loop is checked when the <b>ABNORMAL STATE Field</b> on page 221 is set to yes.
	<b>Alarm Active Unacknowledged</b>	Alarm active and unacknowledged. Select button to acknowledge alarm on CCF or Device faceplates.
	<b>Alarm Active Acknowledged</b>	Alarm is acknowledged and is still active. Button will disappear when alarm returns to normal.
	<b>Alarm Returned to Normal Unacknowledged</b>	Alarm is unacknowledged and has returned to normal. Select button to acknowledge alarm.

## Tool Tips

Tool tips are activated when the cursor is over a tag, description, buttons or an editable field. The tool tip appears as descriptive text on the display.

## Loop FCM Display

The Loop FCM Display, [Figure 13](#), shows the relationships between the functional class modules (FCMs) that make up the loop. Each FCM in the loop is represented by a block on the display.

Both the block type and the name given to the FCM at configuration time appear in the block. Each block contains status information for the FCM.

When a FCM block is selected, the basic control fields become available on the lower part of the display. Use these fields to make mode changes and tune setpoint and output. Runtime versions of the Loop Definition and FCM templates are accessible using the LOOP TMPL and FCM TMPL buttons or by double clicking on the FCM.

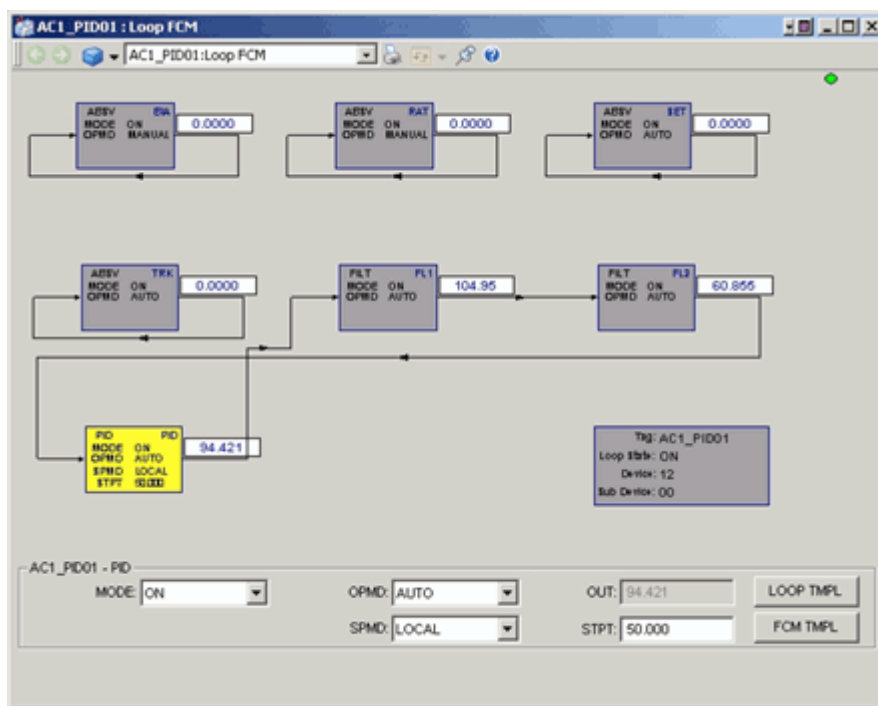


Figure 13. Loop FCM Display

## Operating Procedures for CCF Displays

This section provides information on the following topics:

- [Viewing Trend Data](#)
- [Taking Control of a Loop](#)
- [Changing Control Parameters](#)

### Viewing Trend Data

The source of the values that appear on Loop Detail Display trends are gathered from real-time CCF trend data. Other trend view objects, as described in *Industrial IT 800xA System Operation*, can show real-time and historical trend data for several tags at once.

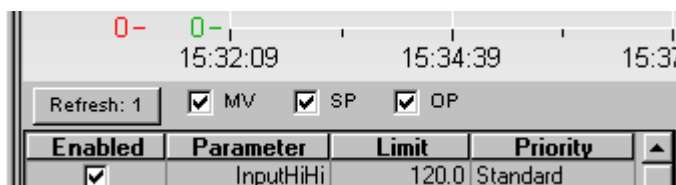
The CCF trend data is scaled, based on the engineering unit range of the tag. When the CCF trend data is available, and the visibility is enabled, CCF trend data is displayed. Real-time trend values are not stored and only exist when the trend is displayed.

#### Working with Loop Detail Display Trend Data

You can change the visibility of trend traces, adjust the refresh rate of the X-axis, the range on the Y-axis and use a ruler to see the numerical values of points on the trend graph.

#### Changing the Trend Refresh Rate

To change trend refresh rate, select the **Refresh** button and enter a value in seconds (1 to 120). A total of 600 samples is shown on a full trend.



### Changing the Trend Visibility

To change trace visibility, select the box next to a trace (see figure above). A checked box becomes a visible trace. Unchecked, a trace is not displayed (data continues to collect).

### Changing the Y-Axis

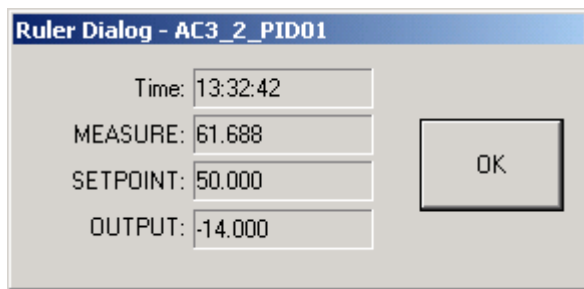
To modify the Y-Axis, select either side of the trend area on the Y-axis markers. This displays the Modify Y-Axis dialog. To change the top number on the Y-axis, select the Top of Range field and enter a new number. To change the bottom number on the Y-axis, select the Bottom of Range field and enter a new number. Click on the **Auto Expand** field to dynamically change the range limits to a new limit value as new values cross the limits. You may also select **Use Output Limits** to dynamically change the range limits to a new limit value as the Output Limits are changed. This only applies to the local machine.



Measure	Setpoint	Output	
Top Of Range	Top Of Range	Top Of Range	OK
100	100	11.4	Cancel
Bottom Of Range	Bottom Of Range	Bottom Of Range	<input checked="" type="checkbox"/> Auto Expand
0	0	-1.4	<input checked="" type="checkbox"/> Use Output Limits

### Showing the X-Axis Ruler

To display the ruler dialog, select a point on the trend graph. A ruler is displayed on the trend graph and moves as the trend is refreshed. This ruler displays the value of points on the trend graph numerically. The Trend dialog displays the numerical Y-axis values where the ruler intersects each trace on the trend graph. You can move the ruler to display the numerical values of other points on the graph by clicking on any other area on the graph. Setpoint applies to PID loop and is zero otherwise. Output applies to control and PID loop and is zero in continuous loops. Select OK to close the Ruler Dialog.



### Taking Control of a Loop

You can take control of a loop from the loop's faceplate or from the Loop FCM display by changing a control parameter. Control parameter changes for a loop tag that are made from these displays allow you to monitor the results of both manual and auto control operations.

The ability of a user to control a tag is determined by the security access rights assigned to the control action for the object. If control access is granted for a tag, you can perform the activities described in the following paragraphs.

### Changing Control Parameters

The method of changing a control parameter for a PID, Control or Device loop is similar in all cases. They may be changed from any operational display where the loop tag target is displayed.

### Using the Faceplate

A general procedure for changing control parameters from a faceplate is provided below.

1. Select the tag target on a graphic or the process values box on a Loop Detail display. This displays the applicable faceplate (based on the kind of tag).

Figure 14 is an example of a faceplate for a PID loop in auto. Faceplates for other types of loops contain information appropriate to the loop type. Measured Value (Pv) is an indicate-only field and cannot be changed.

You may have to adjust some preliminary parameters as a prerequisite for changing the parameter that you actually want to change. For instance, to change the output, the computer mode must be set to local, and the output mode must be set to manual.

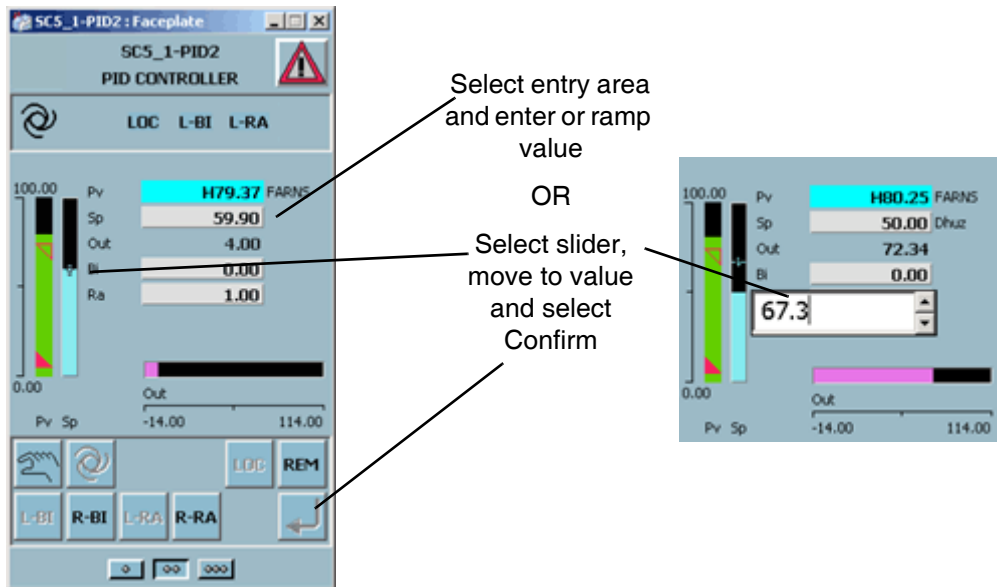


Figure 14. Changing Setpoint on MOD CCF PID Loop Faceplate

2. Select the entry field, and then enter the new value via the keyboard.

3. As an alternative, you can use the control slider to change the setpoint or output value up or down.
4. Select the Confirm button (or press Enter), if enabled, while it is active to accept the new slider value.
5. Repeat the steps above to make changes to the values in any other fields.
6. Click on the **X** button to close the window.

### **Using the Loop FCM Display**

A general procedure for changing control parameters from the Loop FCM display is provided below.

1. Select the FCM on a Loop FCM display.  
Again, you may have to adjust some preliminary parameters as a prerequisite for changing the parameter that you actually want to change.
2. Select the entry field, and then enter the new value via the keyboard.
3. Press Enter to accept the new value.



## Tuning/Engineering Procedures for CCF Displays

This section provides instructions for tuning/engineering functions for CCF displays. These functions are generally restricted to users with greater authority.

Two levels of tuning are supported by the system. The first level is provided by the Loop Detail Display. This display provides the means for basic loop tuning such as gain or limit adjustments. The second level is provided by the Loop FCM Display and runtime versions of the Loop Definition and FCM templets. These displays provide the means for more comprehensive tuning, and for changing loop configurations during runtime (while the system is on-line). Refer to [Engineer Tuning Displays for a Continuous Loop](#) on page 59 for details.

### Tuning Via the Loop Detail Display

You can change tuning, alarm, limits, and loop parameters via the Loop Detail Display. More advanced tuning functions, not supported by the Loop Detail Display, are provided via the Loop FCM Display where you can access the Loop Definition Templet and the Loop FCM Templets for each FCM in the loop.

#### Changing a Tuning Parameter

Tuning parameters include: gain, reset, preact, and adaptive mode. Adaptive mode is a read only field and cannot be changed.

To change a tuning parameter (value entry):

1. Select the Gain value field in the Tuning area of the display. This displays a text entry field within the current Gain field.
2. To change the gain value, enter the new value via the keyboard, and press Enter. The Gain field is updated with the new value.



Gain	0.65
Reset	8.000
Preact	0.000

Changes made to any of the parameters on the Loop Detail Display are not made to the database until you press Enter. Value is reset if focus is lost.



Entered value goes red after it is written to the controller. Once the new value come back from the controller it changes back to black.

### Changing Limit Parameters

Alarm limit parameters include: InputHiHi, InputHi, InputLoLo, and InputLo Measured Variable; Input Rate of Change; and Output Rate of Change. Control limit parameters include: high and low limits for setpoint (SetpointHi, SetpointLo), deviation, and output. Each of these parameters requires a value entry in the Limit column. All conditions in this area can be switched between Enabled (checked) and disabled, and priorities can be set to Standard, High, or Medium.

Quality parameters include: InputBad, SetpointBad, DeviationBad, and OutputBad.

### Changing a Loop Parameter

Loop parameters include: phase, scan rate, loop mode, trend rate, alarm posting, and alarm checking. Phase, scan rate, and trend rate parameters require value entries, the loop mode can be switched between scan ON and scan OFF, and alarm posting and alarm checking can be switched between Enable and Disable.

When changing the trend rate, select one of the following valid choices: 0, 6, 12, 60, 120, 360, 720, 1440, and 2880.

Scan Rate equals the Processing Rate divided by the Base Rate. If the Base Rate is 0.5 and the Processing Rate is 1.0, then the Scan Rate is 2.0. Changing the Scan Rate here actually changes the Processing Rate using the same calculation.

To change a loop mode parameter (list selection):

1. Select the **Loop Mode** value field in the Tuning area of the display. This displays a pull down list to the right of the current field.
2. To change Loop Mode, select the setting you wish to change.

### Changing a Device Parameter

Device parameters include: Simulation Mode, Override Mode, Lock State, and Field State. Field State is a read only field and cannot be changed. The other device parameters can be switched between ON and OFF in the Device Values area.

### Changing Deadband Parameters

Deadband parameters include: Eng Deadband, Output Deadband, and Deviation Deadband. Each of these parameters requires a value entry.

## Engineer Tuning Displays for a Continuous Loop

For tuning and on-line engineering functions that are beyond the scope of the Loop Detail Display, the following set of engineering displays are provided on an individual basis for each loop:

- The Loop FCM Display shows the sources of inputs, the destination of the results, and the status of the FCMs. This display is dynamically updated with runtime data. The display also provides access to the runtime version of the Loop Definition and FCM templets where actual tuning/engineering changes are entered.
- A runtime version of the Loop Definition Templet provides the means to modify some fields that were originally configured via the Loop Definition Templet during control database configuration.
- Runtime versions of the FCM Templets for the loop provide the means to modify some fields that were originally configured via the FCM Templets during control database configuration.

Engineer tuning of continuous loops is initiated by using the Loop FCM Display.

### Loop FCM Display

The Loop FCM Display, [Figure 13](#), shows the relationships between the functional class modules (FCMs) that make up the loop.

Each FCM in the loop is represented by a block on the display. Both the block type and the name given to the FCM at configuration time appear in the block. Each block contains status information for the FCM which can be a combination of the information in [Table 5](#).

The value of the output is displayed on the line that extends from the right side of the FCM block. You can change the values of the status parameters by selecting the block and entering a new value in the fields provided along the bottom of the Loop FCM Display. Continuous values are entered via the regular keyboard keys. For fields that have a predefined list of discrete choices, you can use the pull-down list of valid choices. The new value is sent to the database after the selection is made or when you press **Enter**. The change is only made to the runtime database in the subsystem containing the loop. It is not made to the installed database in the node where the configuration software is resident.

Table 5. FCM Status Information

Label	Description	Possible States
MODE	FCM Mode	ON OFF
OPMD	Output Mode	AUTO MANUAL TRACK (PID and Auto/Manual FCMs only)
SPMD	Setpoint Mode	LOCAL (PID FCMs only) REMOTE (PID FCMs only) TRACK (PID FCMs only)
STPT	Setpoint Value	(PID FCMs only)

The following read-only information for the current loop is provided in the lower right corner of the display:

- Tag Name
- Loop State as specified on the Loop Definition Templet
- Hex address of device/subdevice where the loop resides

To support Loop FCM Display functions, the following links are provided:

**FCM TMPL**           Accesses the runtime versions of the FCM templets for the loop. You must select the box for the desired FCM to see this button before you can select it. You may also double click on the FCM block to get the FCM Templet. See the *Configurable Control Functions (CCF) User’s Guide* for a DETAILED description of the fields on the Loop FCM Templets. See [Appendix A, Runtime Templets](#) for a description of the active runtime fields.

**LOOP TMPL**           Accesses the runtime version of the Loop Definition Templet. [Figure 15](#) shows a typical MOD Loop Definition Templet. See [Appendix A, Runtime Templets](#) for a description of the active runtime fields.

**LINEAR TABLE** Accesses the runtime version of a Breakpoint Sets Templet (Linearization Update Display) for a specific breakpoint set residing on the same node as the current loop. If there are no breakpoint sets residing on the same node as the current loop, a message is displayed and the Linearization Update Display is not called up. If breakpoint sets reside on the node, you are prompted to enter the name of the set you want to modify (not the Templet name). Select the name as created in the NAME OF SET field on the Breakpoint Sets Templet. Selecting any other portion of the Loop FCM Display or entering blanks for the breakpoint set name in response to the prompt aborts the modification process before it begins.

The screenshot shows a window titled "MOD Loop Template - AC1\_PID1". The window contains several fields and controls for configuring a loop template. The fields are organized into two columns.

MOD Loop Template - AC1_PID1		LOOP_DEFINITION (LOOP_DEF)	
TEMPLAT NAME	AC1_PID1		
DEVICE	12		
SUBDEV	0	Save	
LOOP DESCRIPTOR	AIN-PID-AOT CONTRXL LOOP		
PROCESSING RATE	60	LO ENG. UNIT LIMIT	10.
PROCESSING PHASE	3	HI ENG. UNIT LIMIT	200.
NAME OF MEASURED FCM	AIN	MEASUREMENT UNITS	
TREND RATE	0	LOOP STATE	ON
ORDER OF PROCESSING	9999		
SUPPRESS ALARMS	NO	ENABLE RATE ALARMS?	NO
ENABLE POSTING	YES	RATE LIMIT	0
ENABLE CUTOUT	NO	RATE ALARM PRIORITY	STD
CUTOUT SOURCE			

Figure 15. Loop Templet Display

### Runtime Versions of Loop Definition and FCM Templets

The runtime versions of the templets are used to make the actual tuning changes. The templets are accessed via the Loop FCM Display as described above. Their format is similar to the configurator templets in AdvaBuild with the following exceptions:

- The information is current since it comes directly from the active database.
- Black values are tunable while the grayed values are not.
- The AdvaBuild versions of the templets have different buttons.
- No FCM list is on the runtime version of the Loop Definition Templet.

Tuning changes are made to database attributes by first changing the information on the templet and then saving the templet information to the database by selecting the **SAVE** button. If the save process is not successful, a message describing the problem is displayed. After corrective action is taken, you can re-attempt the save process. Use the **X** button to exit the templet.



Tuning changes affect both the active database in the subsystem containing the loop and the installed database on the disk in the node where the AdvaBuild configuration software resides. Matching the installed database with the operational database allows tuning changes to remain in effect when a subsystem or control module is rebooted.

Templets originally used to configure the database retain the old value until the database is decompiled. Instructions for decompiling a database are provided in *AdvaBuild Basic Functions User's Guide* for the Advant Station-based AdvaBuild software.

### Runtime Version of Breakpoint Sets Templet

The runtime version of the Breakpoint Sets Templet (Linearization Update Display) is used to change the values of a specific breakpoint set without having to recompile and install the entire CCF configuration of the node. The templet is accessed via the Loop FCM Display as described above. Its format is similar to the configurator templet with the following exceptions:

- Only the X VALUE Y VALUE edit window with a blue background is tunable.

- The buttons are LOOP DISPLY and SAVE. Selection of the LOOP DISPLY button prior to saving any modifications you made returns you to the Loop FCM Display without saving the modifications.

Tuning changes are made to the database by first changing the values in the edit window and then sending the new values to the database. The values are changed using the same methods you used to define them. That is, enter up to 11 x,y pairs in the edit window.

When all required changes to the values are made, the new values are sent to the database by selecting the SAVE button. If the save process is not successful, a message describing the problem is displayed. After corrective action is taken (this usually involves entering another x,y pair since you must define at least two x,y pairs), you can re-attempt to save the new values.

This functionality only allows you to modify linearization tables existing on the same node as the current loop. It does not allow you to modify linearization tables residing on another node, create new tables, delete existing tables, or change the name of the breakpoint set table.

The procedure described above changes both the active database in the subsystem containing the loop and the installed database on the disk in the node where the configuration software resides. However, the Breakpoint Sets Templets originally used to configure the database is not immediately modified. It retains the old values until the database is decompiled. Instructions for decompiling a database are provided in *AdvaBuild Basic Functions User's Guide* for the Advant Station-based AdvaBuild software.

There is no log of the changes made to the breakpoint sets.

## Engineer Tuning Displays for a Device Loop

Access the runtime version of the Device Loop Templet by:

1. Select the Loop FCM Display.
2. Select the device block.
3. Select LOOP TMPL.

There is no FCM Templet for device loops. Use the fields at the bottom of the Loop FCM Display to make State, Mode and Command changes.

## Alarms

Alarms based on the process measurement are defined as part of the loop. Alarms on Setpoint, Output, Deviation, or other control parameters are defined as part of the FCM. Each alarm has several parameters that can be set on an alarm type basis. Each alarm type can be turned on or off. The assigned priority to the alarm type is set to STANDARD, MED, or HIGH.

### Alarm Post and Alarm Check

Alarm suppression is a useful feature during startup or shutdown conditions when nuisance alarms can be generated by equipment that is not in service. ALARM CHECK can be turned off (Disabled) during periods when the alarm conditions should not be tested. ALARM POST provides the capability to prevent (Disabled) alarms from printing and/or generating audible indication on operator consoles. With ALARM CHECK Enabled and ALARM POST Disabled, the alarm conditions will be tested and will be visible on displays; but, they will not generate unwanted audible indications.



MV	58.4543	FARNS
SP	50.0000	LOCAL
OP	H101.0306	AUTO
Phase	1	
Scan Rate	2	
Loop Mode	ON	
Trend Rate	60	
Alarm Post	Enabled	
Alarm Check	Enabled	

Use pull-down selections on the Loop Detail display to Enable/Disable Alarm Post and Alarm Check functions.

### Alarm Detection

PID loops detect alarms on the measured variable, setpoint, deviations, and output. Control loops (Auto/Manual for example) detect alarms on the measured variable and output. Continuous loops (indicator and calculator for example) detect alarms on the measured variable.

Measured variable alarms are trip points. Setpoint and output alarms are limit alarms on their high or low value. Deviation alarms compare the difference between the measured variable (going into the PID FCM) and the setpoint. A high deviation occurs when the measured variable of the PID FCM is above the setpoint by the limit amount. The deviation value is never limited by the system.



The input and output rate alarms compare the number of engineering units set as the limit to the actual change in one loop cycle.

Discrete loops detect device related alarms. The device alarms are:

- **ILLEGAL** (Illegal State or Command). The system can read the feedback from the device, but no state is defined for the feedback bit combination or Systems recognize the command from the operator or the TCL program, but it is an invalid command as defined by the Device Descriptor Templet.
- **ABNORMAL** (Abnormal State Change). The device changed state, but no command to do so was issued. Abnormal State Change alarms will not occur if the state changes to match DEV\_CMND.
- **TRANSITION** (Unknown State or Timeout). The system cannot read the feedback from the device or the system sent a valid command to the device, but the device did not respond within the specified time.
- **STATE** (State Alarm). The device has entered a state defined by the device descriptor set as an alarm state for the device.

## Alarm Indication and Viewing

Alarm priority is indicated using colored backgrounds and the characters H (high), L (low), D (Deviation), R (Rate) and ? (Bad data quality) in front of the display value. The background color of the highest priority alarm currently active is displayed. An unacknowledged alarm blinks. On faceplates, an Alarm Acknowledge button appears when an unacknowledged alarm exists. Indications are also made for Active Alarm Acknowledged and Inactive Alarm Unacknowledged.

Alarm conditions may also be indicated audibly by a .WAV file. Various combinations of audible and visual alarm indication can be obtained by configuration. See *Industrial IT 800xA System Operation* for a description of the event bar and event display included with the base product.

## Alarm Priority

A priority (High, Medium, Standard) can be assigned to each alarm, [Figure 16](#). Alarm priorities are used to display a more important alarm before a less important alarm when viewing the alarm displays.

Enabled	Parameter	Limit	Priority	
<input checked="" type="checkbox"/>	InputHiHi	100.0	High	
<input checked="" type="checkbox"/>	InputHi	100.0	Medium	
<input checked="" type="checkbox"/>	InputLo	5.0	Standard	

Figure 16. Alarm Limit and Priority Settings



If you are using both the deviation and setpoint limit alarms, it is possible, when the alarms are not acknowledged and priorities are the same, to see the deviation active on the faceplate and the setpoint limit active on the loop detail. Use different priorities.

## Alarm Limits and Limiting

The high and low alarm limit trip-points for each control loop are determined at the time of configuration and can be modified during operation, [Figure 16](#).

Process and deviation alarms are triggered by the process input. A process alarm trips (becomes active) when the process reaches a preset high or low trip-point. A deviation alarm trips when the process value deviates from the control loop setpoint by a preset amount (deviation high value is when the measured value into the PID FCM is larger than the setpoint and deviation low value is when it is less than the setpoint).

An output alarm is activated when the control loop output reaches a preset high or low limit. Alarms resulting from rising values are defined as high, and those resulting from falling values are defined as low. The terms high high and low low mean that two separate alarms are configured on a single variable with either two high or two low trip points.

The high, low, and rate of change alarms are extensions of output limiting. For example, the OUTPUT HIGH LIMIT specifies the maximum output value. This value is used as a limit in modes where limiting is allowed by the LIMITED OUTPUT MODES field. In a similar manner, the high and low setpoint alarms can be limited by the LIMITED SETPT MODES field. Ratio and bias are limited by the LIMITED RATIO MODES and LIMITED BIAS MODES fields.

## Alarm Acknowledgement

Alarms provide indication on each unacknowledged and active alarm. You can acknowledge all alarms for a tag at once using either the Acknowledge Alarm button or by selecting the context menu for the object and using **Acknowledge** verb. Alarm acknowledgement may also be done through Alarm lists and graphics. [Figure 17](#) illustrates Alarm Acknowledgement from the faceplate.

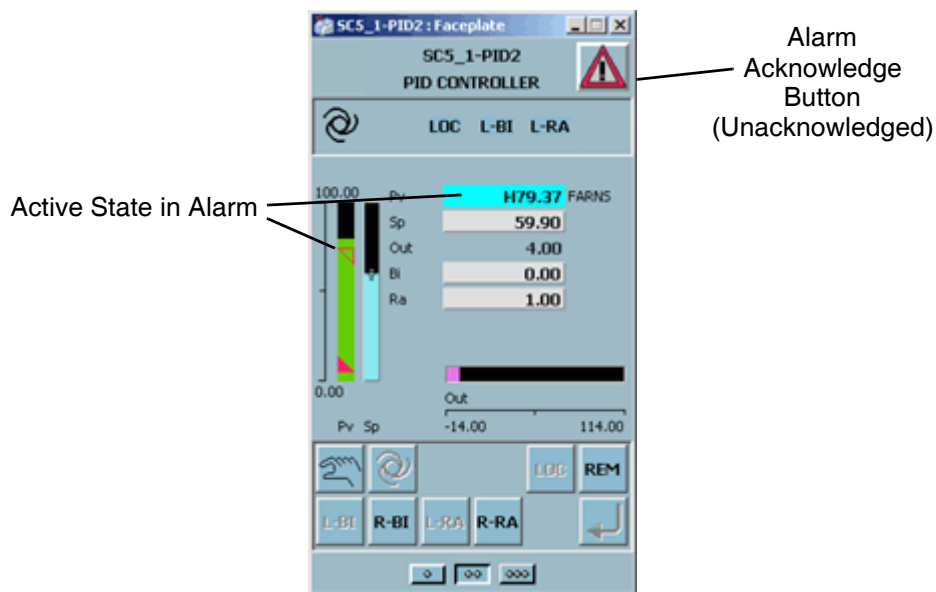


Figure 17. Alarm Acknowledgement on Faceplate

## MOD Group Displays

A group is a collection of related loops. Generally, a group has up to 12 control loops or up to 36 indicator loops or a combination of both. Control loops assigned to indicator targets provide limited information and cannot be used for control.

Monitoring and control functions (control is enabled in the MOD Group Object) are provided in the Group Status and Group Trend displays. A custom graphic can be configured for the Group Graphic display. The Group Alarm display is a filtered list of a selected event group page.

## Group Status Display

The Group Status Display, [Figure 18](#), shows the current status of all tags within the group by organizing the tags into status blocks (one per control loop, and up to three indicator loops). Trend data is shown for PID and A/M control loops which have trending configured in the database. Device loop status blocks show tag, descriptor, command state, mode, and state. Indicator loop status blocks show basic information for up to three tags.

SCS_2-1:MOD300 Group Status			
SC5_2-ADD1 SUM OF 4 INPUTS 62.2860 RQAGS	SC5_2-PID1 TC100 SIMULATION LOOP 5.0000 DEGF	SC5_2-DIV1 DIVISION 0.0000 JOTS	SC5_2-DTM1 DEAD TIME 0.0000 XEODS
SC5_2-AI1 ANALOG INPUT 0.0000 GLEEBBS	SC5_2-NPT1 NON-RATE PERIODIC TOTAL 6.1490 ZORKS		
SC5_2-AND1 LOGICAL AND 0.0000 BUGS	SC5_2-CNT1 COUNTER -1.747e+006 PHIPS		
SC5_2-FF1 FLIP-FLOP 0.0000 ABS	SC5_2-INT1 TIME INTEGRATION 6.421e+006 PUOIDS	SC5_2-AVG1 AVERAGE 45.2858 GOBS	SC5_2-PMX1 PERIODIC MAXIMUM 6.1490 BINS
		SC5_2-TOT1 TOTALIZER 120.0000 WONGS	SC5_2-MTH1 MATH 0.0000 PINGS
		SC5_2-PTO1 PERIODIC RATE TOTAL 1866.2397 ZEKES	SC5_2-DER1 TIME DERIVATIVE -0.0192 NOBS
SC5_2-DTC1 DEAD TIME COMPENSATION 0.0000 LERGS	SC5_2-EXP1 EXPONENTIATION 0.0000 KLEETS	SC5_2-FLT1 FIRST ORDER FILTER 55.9815 KOUDS	SC5_2-MUL1 MULTIPLICATION 0.0000 MUNGS

Figure 18. Group Status Display

### PID and A/M Status Blocks

The PID status block shows tag, descriptor, setpoint (SP) value and mode, measured variable value, engineering units, output value and mode, computer mode, a current output value bar graph, a current measured value bar graph with setpoint and limits, engineering low limit, engineering high limit, FF/FB, and a trend graph. Alarm indication and viewing is as described in [Alarms](#) on page 64. The Auto Manual and

continuous loops are similar to the PID minus the setpoint and FF/FB mode information.

The trend graph represents one division of trended data (60 sampled points) from the engineering low limit to the engineering high limit. The most current trend sample is on the right side of the graph. Trend data is only displayed if the trend rate is enabled (not 0) for the tag. To see the trend data for a specific time, select on the trend graph. A trend ruler with the measured value, time and data quality for that data point is then shown. To view fully functional trends for a group, access the Group Trend Display.

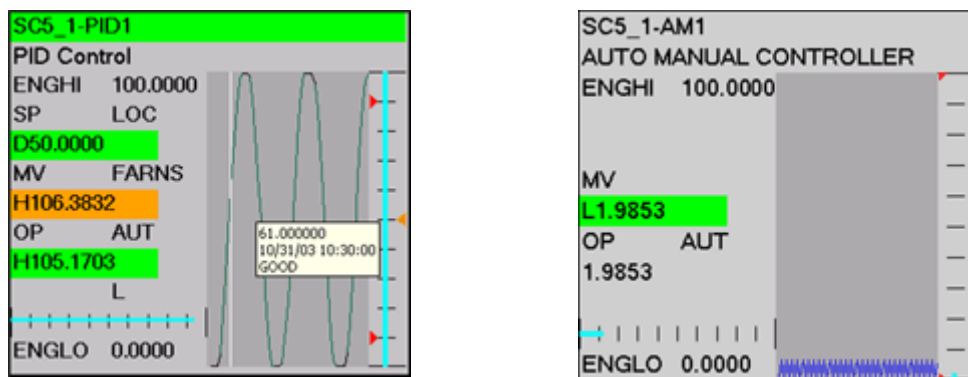


Figure 19. PID and A/M Status Blocks

Control functions within the block depend on how the loop is configured and user authority. Setpoint mode and value, computer mode, output mode and value, and FF/FB are all controllable. The block is context sensitive to its own tag. Double clicking on the tag name displays the loop faceplate.

### Continuous Loop Status Blocks

Continuous loop status blocks show tag name, description, measured value and engineering units. There are no control items.

### Device Loop Status Blocks

Device loop status blocks show tag, descriptor, command state, mode, and state. The command, mode, and state values can be modified if enabled for control in the MOD Group Object. Only device descriptor command and state values of 0 to 15 are shown. Use the loop fcm if your configuration uses greater values.

### Indicator Status Blocks

Indicator loop status blocks show tag and descriptor as well as measured variable, engineering units and alarm indications for PID, continuous or A/M loops or command state for device loops. There are no control elements for tags using the indicator loop status block.

### Alarm Indication and Viewing

Alarm indication and viewing is as described in [Alarms](#) on page 64.

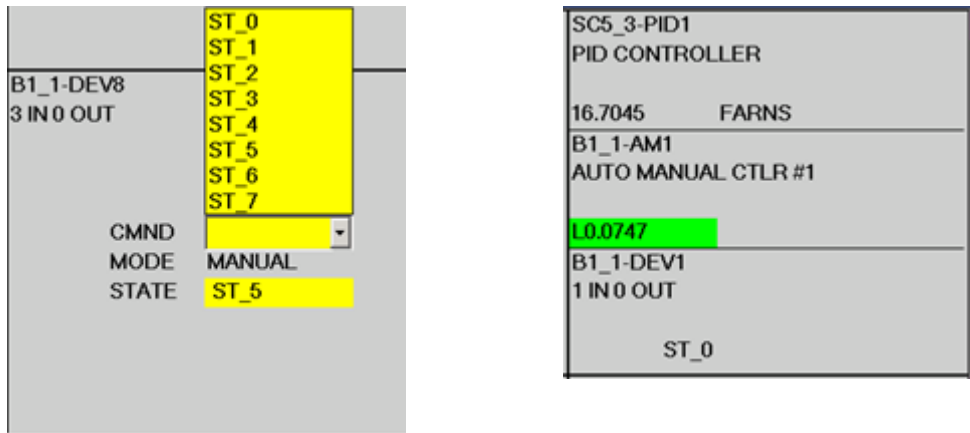


Figure 20. Device Loop and Indicator Status Blocks

## Group Trend Display

The Group Trend Display, [Figure 21](#), provides trend records for up to 12 loops. Group trends are displayed on three full-width graphs. Each graph shows trends for up to four loops, and is accompanied by the corresponding status blocks. The most current trend sample is on the right side of the grid. MOD 300 trend buffers are supported and the most recent 300 points are read if a Log Configuration aspect is configured for the tag.

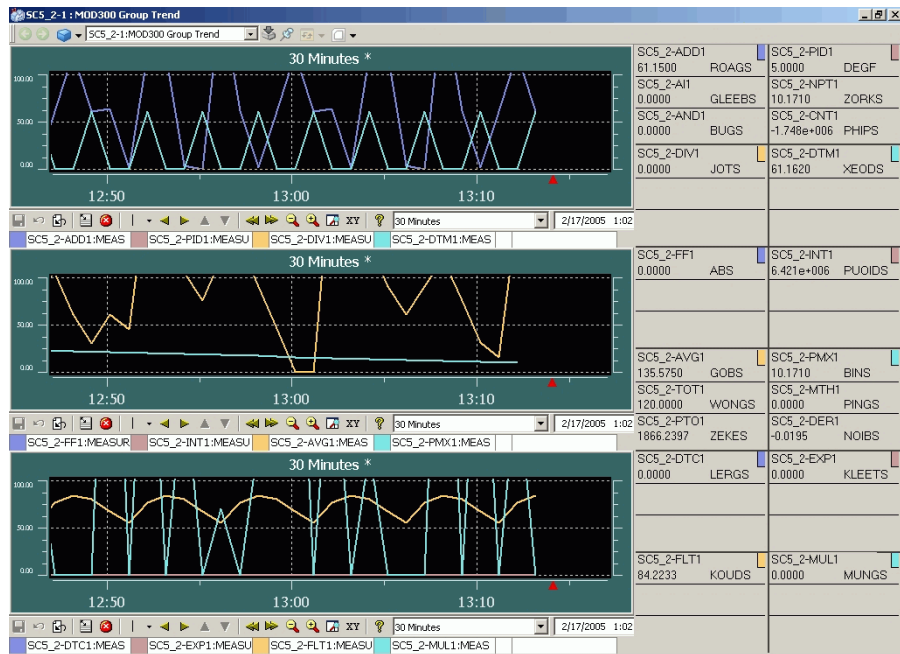


Figure 21. Group Trend Display

### Group Trend Graph

Each of the four tags on a graph can have its own vertical scale (if limits are different) with the upper and lower ranges displayed next to the trend graph. A maximum of 300 data points can be trended for each tag. To display the value of a point on a trend graph numerically, click on a trend trace.

Each trend line in a graph is color coded to match its tag and range color. The context menu for the trend is the same as for the standard trend.

**Group Trend Status Blocks**

To make changes to a loop on the Group Trend Display, select the desired field in the loop’s status block, [Figure 22](#). Control functions within the block depend on how the loop is configured and user authority. Setpoint mode and value, computer mode, FF/FB, output mode and value are all controllable. The block is context sensitive to its own tag. Double clicking on the tag name displays the loop faceplate.

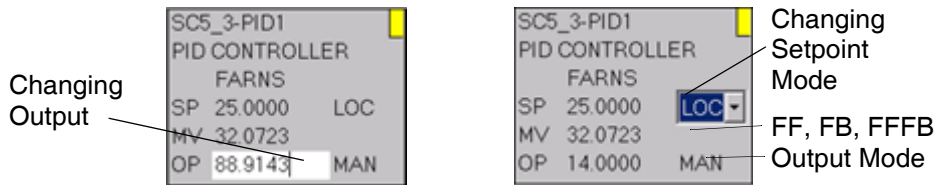


Figure 22. Changing Values in Status Blocks for Group Trends

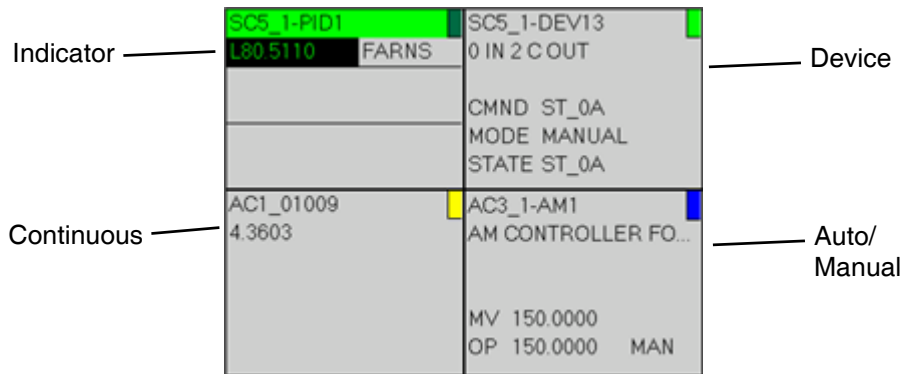


Figure 23. Sample Group Trend Status Blocks

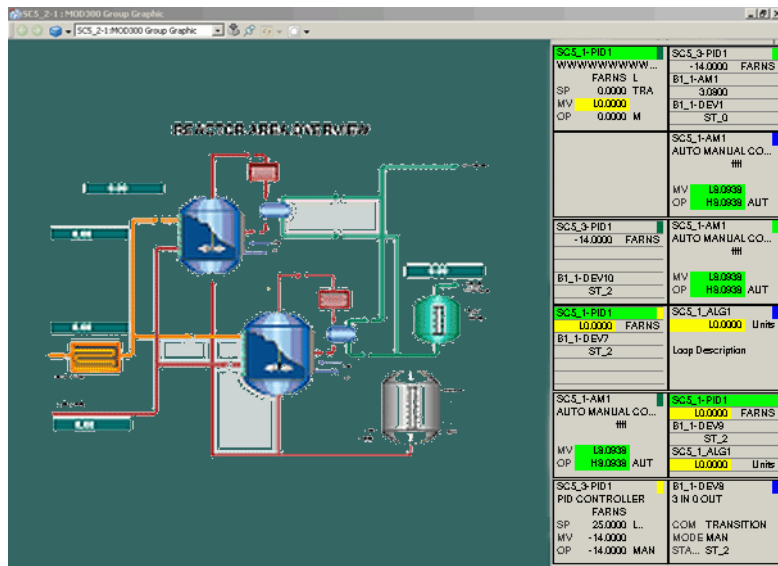
**Alarm Indication and Viewing**

Alarm indication and viewing is as described in [Alarms](#) on page 64.



## Group Graphic Display

The Group Graphic Display, [Figure 24](#), presents a custom graphic for a group and status blocks for each tag in the group. The status blocks occupy the right one-third of the screen, and the graphic occupies the left two-thirds of the screen. Status blocks operate as described in [Group Trend Status Blocks](#) on page 72.



*Figure 24. Group Graphic Display*

Typically, tags in the group are represented by one or more graphic symbols on the graphic display. The symbols may be static or dynamic (changing to show dynamic changes in the process).

There is a Group Graphic aspect for each group. The graphic configured for that aspect gets displayed in the Group Graphic display.

## Group Alarm Display

The Group Alarm Display is a selected event group page with an additional filter applied based upon the tags assigned to the MOD environment group. The primary event group page is a filtered set of event attributes such as state (acknowledged and active) and type (operator action and process related). All events matching the event filter are considered to be part of the event group.

The filters defined in the event group narrow the scope of events seen by the event page. The event page can provide further filtering of the events but cannot expand beyond the filter of the event group.

When the Group Alarm display is first called up, the primary filter is applied and then the filtered list of tags assigned to the group is applied.

## MOD Area Displays

An area is a collection of up to three groups. As such, each area is composed of up to 108 loops (control loops, indicator loops, or a combination of both). This provides a broader view of the process from a single display. Each area supports monitoring and control functions using Area Alarm and Area Status displays. A custom graphic display can be configured for the Area Graphic display.

### Area Status Display

The Area Status Display, [Figure 25](#), shows the current status of all tags in the area (up to three groups).

The display is divided by rows into three groups with Group 1 in the top third, Group 2 in the middle third, and Group 3 in the bottom third. As many as 12 status lines make up a group.

Tag Name	Value	Unit	Control	Status	Tag Name	Value	Unit	Status	Tag Name	Value	Unit	Status
SC5_2ADD1	122.3320	ROAGS	SC5_2AI1	0.0000	GLEEBS	SC5_2AND1	0.0000	BUGS				
SC5_2PID1	5.0000	DEGF	SC5_2NPT1	2.1660	ZDRKS	SC5_2CNT1	-1.747e+006	PHIPS				
SC5_2OIV1	0.0000	JOTS										
SC5_2DTM1	61.1530	XEODS										
SC5_2FF1	0.0000	ABS										
SC5_2INT1	6.421e+006	PUIODS										
SC5_2AVG1	91.1660	GOBS	SC5_2TOT1	120.0000	WONGS	SC5_2PTO1	1866.2397	ZEKES				
SC5_2PMX1	2.1660	BINS	SC5_2MTH1	1.0000	PINGS	SC5_2DER1	0.0194	NOIBS				
SC5_2DTC1	0.0130	LERGS										
SC5_2EXP1	1.0000	KLEETS										
SC5_2FLT1	84.2314	KDUDS										
SC5_2MUL1	351.1595	MUNGS										
SC5_2TB6-1	TEST LOOP 1	AUTO	OFF	'NONE*****								
SC5_2TB6-2	TEST LOOP 2	AUTO	OFF	'NONE*****								
SC5_2TB6-3	TEST LOOP 2	AUTO	OFF	'NONE*****								
SC5_2TB6-4	TEST LOOP 2	AUTO	OFF	'NONE*****								
SC5_2NFM1	44.0237	SERDTS	SC5_2FMN1	0.0000	SCRODS	SC5_2PUT1	0.0000	ULMS				
SC5_2RED1	0.0000	NORGS	SC5_2SOR1	436.9995	GRUBS	SC5_2TMP1	0.7337	HOONS				
SC5_2MTH3	1.0000	PINGS										
SC5_2MTH5	1.0000	PINGS										
SC5_2MTH7	1.0000	PINGS										
SC5_2MTH9	1.0000	PINGS										
SC5_2PAV1	1.0830	WAGS	SC5_2PSD1	1.8204	GERTUN	SC5_2RB1	121.1660	PLOMS				
SC5_2SEL1	0.0000	FRAGS	SC5_2TIM1	1.1660	WEEZES	SC5_2MTH2	1.0000	PINGS				
SC5_2MTH4	1.0000	PINGS										
SC5_2MTH6	1.0000	PINGS										
SC5_2MTH8	1.0000	PINGS										
SC5_2MTH10	1.0000	PINGS										

Figure 25. Area Status Display

One to three indicators are displayed on one status line, [Figure 26](#), depending on the configuration. Each indicator line segment is context sensitive to its own tag.

Tag	Measured Variable Value	Eng Units	Tag	Measured Variable Value	Tag	Device State
AC1-PID1	50.5000	GPM	SC5_1_ALG1	L5.0000	B1_1-DEV1	ST_0

Figure 26. Indicate Status Line (PID, Continuous, Device)

For a PID control loop, information is displayed on one status line, [Figure 27](#). The measured value bar graph also indicates the setpoint value with an orange triangle on the bottom of the graph, and it indicates the input high and low limits with red triangles on the top of the graph.

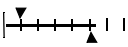
Tag	Descriptor	Setpoint Mode	Setpoint Value	Measured Variable Value	Eng Units	Cmp Mode	Bar Graph EngLo - EngHi	FF/FB Mode	Output Mode	Output Value
AC1-PID1	Unit 1 Controller	REM	50.0000	50.5000	GPM	L		FF	AUT	25.0000

Figure 27. PID Status Line

The target areas within the status line depend on how the loop is configured and user authority. Controllable items are setpoint mode and value, computer mode, FF/FB mode, output mode and value. The line is context sensitive to its own tag. Double clicking on the tag displays the loop faceplate. Auto manual loops are similar to the PID status line without the PID specific information.

The Device Loop status line includes one device loop tag, [Figure 28](#). The line is context sensitive to its own tag. Mode and Command are controllable depending upon authority.

Tag	Descriptor	Mode	Command	State
B1_1-DEV8	3 IN 0 OUT	AUTO	TRANSITION	STATE 3

Figure 28. Device Status Line

## Area Alarm Display

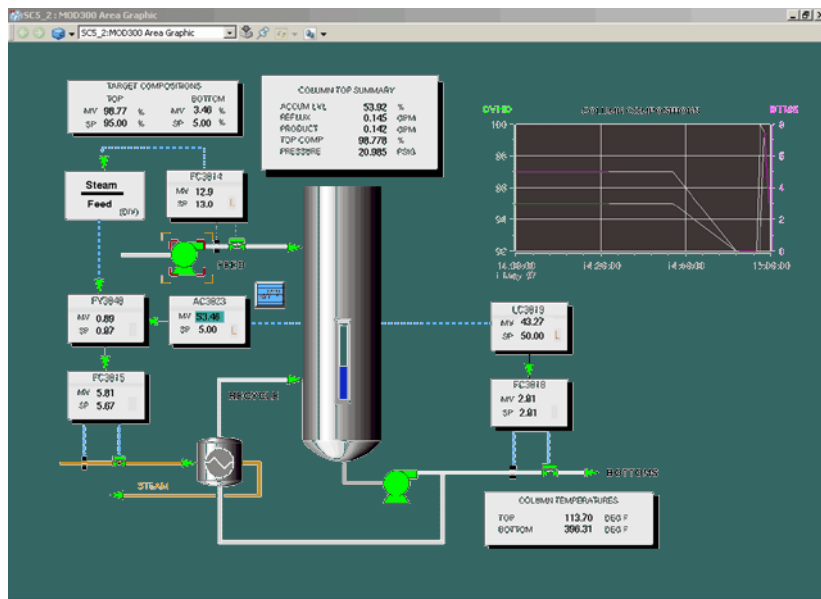
The Area Alarm Display is a selected event group page with an additional filter applied based upon the tags assigned to the MOD environment area. The primary event group page is a filtered set of event attributes such as state (acknowledged and active) and type (operator action and process related). All events matching the event filter are considered to be part of the event group.

The filters defined in the event group narrow the scope of events seen by the page. The page can provide further filtering of the events but cannot expand beyond the filter of the event group.

When the Area Alarm display is first called up, the primary filter is applied and then the filtered list of tags assigned to the area is applied.

## Area Graphic Display

The Area Graphic Display, [Figure 29](#), provides a custom graphical depiction of a given area. Any tag in the area may be represented by one or more graphic symbols on the graphic display. The symbols may be static or dynamic (changing to show dynamic changes in the process). The graphic symbols and their dynamics are defined during display building. The Area Graphic Display does not have a standard format. These displays are custom designed to meet the process monitoring and control requirements of a specific application.



*Figure 29. Area Graphic Display*

There is an Area Graphic aspect for each area. The graphic configured for that aspect gets displayed in the Area Graphic display.

---

## Section 4 TCL Displays

### Operational Displays for TCL

This section describes the standard operational displays for sequential control using the Taylor Control Language (TCL). Sequential control operational displays support monitoring and control functions for TCL. Functions you can perform using the sequential operational displays include:

- Monitoring units and sequences
- Activating and deactivating programs
- Controlling program state, status, and mode
- Manipulating steps
- Changing recipe values
- Changing tag parameters

Users with the proper access authority can also troubleshoot and debug programs under runtime conditions from a Sequence Debug Display as well as receive and respond to unit messages. Users who are not assigned to a particular unit can monitor programs on the unit; however, they cannot perform control functions or receive messages.



The Acknowledge selection on the TCL context menu acknowledges all the messages from any of the TCL displays.

### Displays for Sequence and Recipe Development

TCL sequences and recipes are developed using the TCL Editor and TCL Recipe Editor respectively. See [Related Documentation](#) on page 21.

## TCL Version Mismatch Warning

If the source code of a TCL sequence is re-compiled or modified after that sequence has been loaded, a warning message, [Figure 30](#), is displayed on the Unit Detail, SFC, Sequence Detail, or Sequence Debug display as follows:

WARNING: SEQUENCE SOURCE MISMATCH

On the Unit Detail page, the affected sequences are also indicated by the Sequence ID having yellow text.

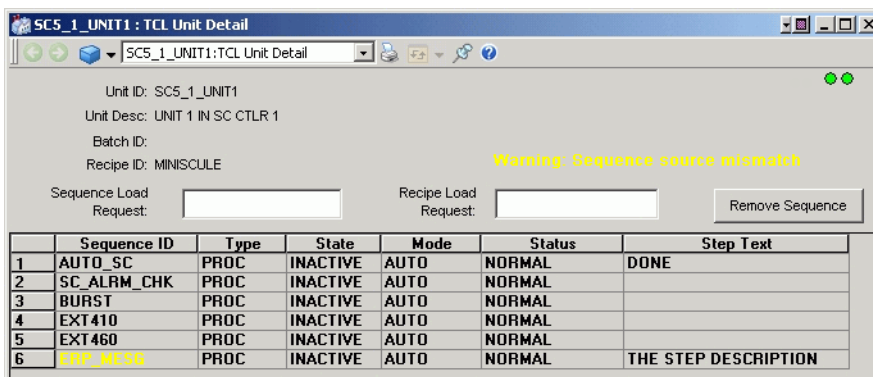


Figure 30. TCL Version Mismatch Warning

## TCL / Unit Message Display

The TCL / Unit Message Display, [Figure 31](#), lists certain need-to-know process condition messages sent from executing TCL programs. Messages which require a response cause the originating sequence to pause until an appropriate response is entered.

The TCL Message Display includes all TCL messages while the Unit Message Display includes only those TCL messages associated with the selected unit. These message displays are similar to the standard alarm/message displays as described in the *Industrial IT 800xA System Operation* book for the base product.



## TCL Message Display Format

The data displayed for each message includes the acknowledgement status, colored priority/alarm indication (High, Medium, Standard), event type, arrival date and time, the unit from which the message was sent and the message itself.

## Acknowledge and Respond to a Message

Acknowledge individual events by selecting the Ack box or a group of events by selecting a group of boxes. Acknowledge a page of messages by using **Acknowledge** on the menu line.

To respond to a TCL message, select the appropriate line (MOD TCL Reply category) and enter the appropriate answer in the reply field, [Figure 31](#), and OK the message. Messages requiring a response are removed from the screen after a response is given.

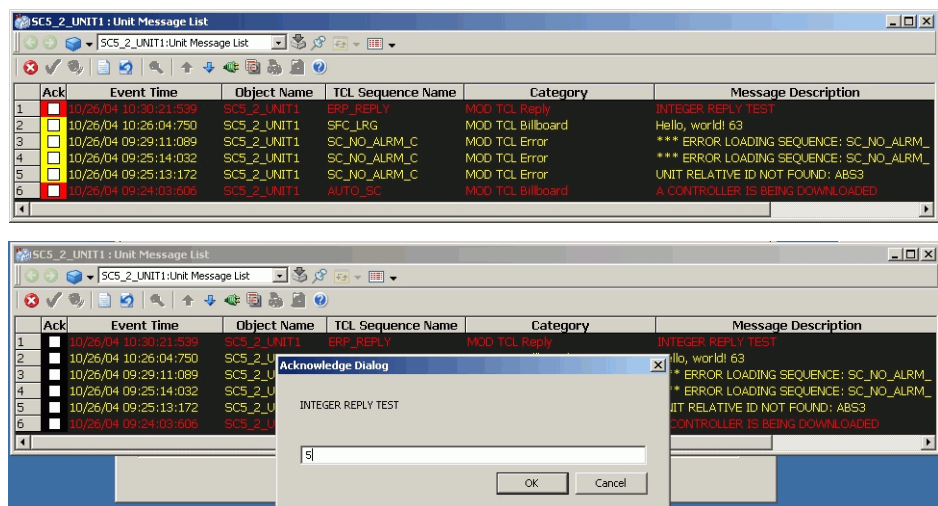


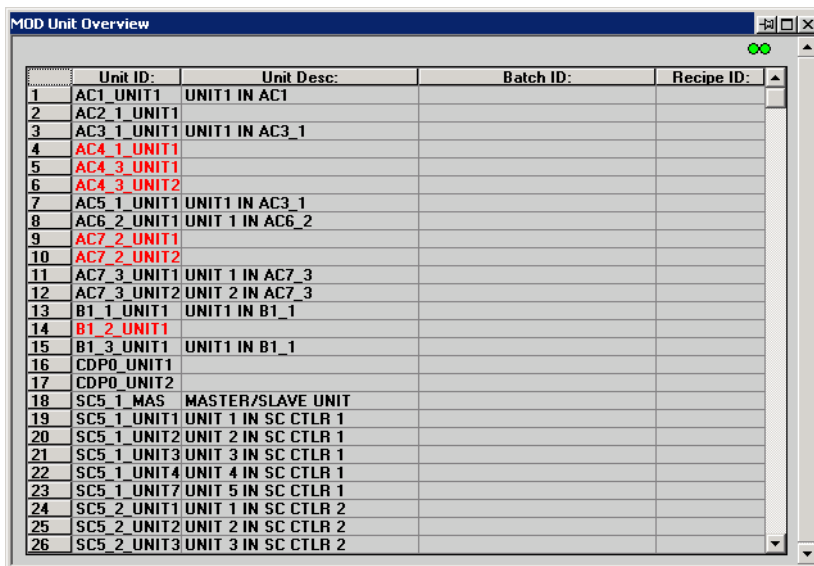
Figure 31. TCL Message Display

## Unit Overview Display

The Unit Overview Display, [Figure 32](#), lists all the units configured for the system, as well as the batches and recipes associated with each unit. Units are sets of devices and their associated tanks, reactors, and so on which are treated as entities when being controlled by TCL programs. Display access is from the context menu of the object browser. The Unit Overview Display provides the following information for each unit:

- Unit ID (name) and description
- Batch ID, if any
- Recipe ID currently associated with the unit, if any

Click on the Unit ID field to access the Unit Detail display for the selected unit.



	Unit ID:	Unit Desc:	Batch ID:	Recipe ID:
1	AC1_UNIT1	UNIT1 IN AC1		
2	AC2_1_UNIT1			
3	AC3_1_UNIT1	UNIT1 IN AC3_1		
4	AC4_1_UNIT1			
5	AC4_3_UNIT1			
6	AC4_3_UNIT2			
7	AC5_1_UNIT1	UNIT1 IN AC3_1		
8	AC6_2_UNIT1	UNIT 1 IN AC6_2		
9	AC7_2_UNIT1			
10	AC7_2_UNIT2			
11	AC7_3_UNIT1	UNIT 1 IN AC7_3		
12	AC7_3_UNIT2	UNIT 2 IN AC7_3		
13	B1_1_UNIT1	UNIT1 IN B1_1		
14	B1_2_UNIT1			
15	B1_3_UNIT1	UNIT1 IN B1_1		
16	CDP0_UNIT1			
17	CDP0_UNIT2			
18	SC5_1_MAS	MASTER/SLAVE UNIT		
19	SC5_1_UNIT1	UNIT 1 IN SC CTLR 1		
20	SC5_1_UNIT2	UNIT 2 IN SC CTLR 1		
21	SC5_1_UNIT3	UNIT 3 IN SC CTLR 1		
22	SC5_1_UNIT4	UNIT 4 IN SC CTLR 1		
23	SC5_1_UNIT7	UNIT 5 IN SC CTLR 1		
24	SC5_2_UNIT1	UNIT 1 IN SC CTLR 2		
25	SC5_2_UNIT2	UNIT 2 IN SC CTLR 2		
26	SC5_2_UNIT3	UNIT 3 IN SC CTLR 2		

Figure 32. Unit Overview Display

## Unit Detail Display

The Unit Detail Display, [Figure 33](#), lists the sequences that reside on and can be run on a unit. This display is used to:

- Load a sequence onto the unit
- Load a recipe onto the unit
- Make state, mode, and status changes
- Remove a sequence from the unit when it is no longer needed
- Select a specific Sequence Detail, Debug or SFC display

	Sequence ID	Type	State	Mode	Status	Step Text
1	AUTO_SC	PROC	INACTIVE	AUTO	NORMAL	DONE
2	SC_ALARM_CHK	PROC	INACTIVE	AUTO	NORMAL	
3	BURST	PROC	INACTIVE	AUTO	NORMAL	
4	EXT410	PROC	INACTIVE	AUTO	NORMAL	
5	EXT460	PROC	INACTIVE	AUTO	NORMAL	
6	ERP_MESG	PROC	INACTIVE	AUTO	NORMAL	THE STEP DESCRIPTION

Figure 33. Unit Detail Display

## Unit Detail Display Format

The top of the display contains the unit ID and description, the ID of the recipe associated with the unit, and the batch ID. There is a field for entering a sequence load request, and also a field for entering a recipe load request.

The information shown in the main portion of the display for each sequence in the unit includes:

Sequence ID	Sequence name as listed on TCL Catalog
Sequence Type	Schedule, Procedure, or Operation
Sequence State	Active, Inactive, Paused, Loading
Sequence Mode	Auto, Manual, Semi_Auto
Sequence Status	Normal, Abnormal 1 - 16
Step Text	Current step in active sequence or if inactive, step executed when sequence became inactive

Each column width can be adjusted by clicking on the title border and sliding the width indicator left or right.

## Loading the Sequence from the Unit Detail Display

Note the ID of the sequence you want to download (refer to your AdvaBuild documentation). Access the Unit Detail Display of the unit where the sequence is to be downloaded.

To load a sequence onto a unit:

1. Select the Sequence Load Request field
2. Enter the ID for the desired sequence (optionally, you may load the sequence to a slave unit by entering , **SLAVE** after the Sequence ID)
3. Press **ENTER**

Wait while the system downloads the sequence to the unit. The message

Loading Sequence *sequence name*

is displayed while the system downloads the sequence. When downloading is complete, the sequence and its related data are shown in the list of sequences. The initial state of a just-downloaded sequence is inactive.

## Loading a Recipe from the Unit Detail Display

Note the ID of the recipe you want to download (refer to your Advabuild documentation). Access the Unit Detail Display of the unit where the recipe is to be downloaded.

To load a recipe onto a unit,

1. Select the entry field to the right of the **Recipe Load Request** in the Unit Detail Display.
2. Type the ID for the desired recipe (optionally, you may load the recipe to a slave unit by entering **, SLAVE** after the recipe ID)
3. Press **ENTER**. The message

`Loading Recipe recipe name`

is displayed while the system is downloading the recipe.

When downloading is complete, the new Recipe ID is displayed in the Recipe ID field. You can load one recipe per unit.

## Changing Sequence State, Mode, and Status

The sequence state, mode, and status can be changed by selecting one of the sequences listed on the Unit Detail Display. Use the pull down boxes to make your selection.

Functions for sequence state are:

ACTIVATE	Starts executing steps.
RESUME	Starts a paused sequence.
PAUSE	Stops a sequence between steps, making it paused.
ABORT	Ends a sequence in the middle of a step, making it inactive.

For example, to activate an inactive sequence, select **ACTIVATE**. To end an active sequence, select **ABORT**. The abort action causes the TCL sequence to end immediately without further action by that sequence while the current process conditions remain unchanged.

Functions for sequence mode are:

- AUTO**                      Programs execute steps in order without requiring operator intervention.
- SEMI\_AUTO**              Programs execute steps automatically, but pause at configured breakpoints. The operator can then resume or abort.
- MANUAL**                   Programs execute manually, one step at a time.

The semiautomatic mode and manual mode support troubleshooting and debugging sequences.

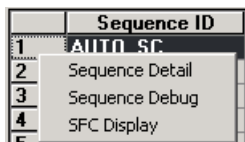
Functions for sequence status are:

- NORMAL**                   Indicates that no abnormal conditions exist. Normal is the default mode after downloading.
- ABNORMAL**               Indicates that an abnormal condition exists. The number to the right of **ABNORMAL** defines the type of condition. See *Taylor Control Language (TCL) for AdvaBuild User's Guide* for information about the abnormal conditions.

## Removing one or more Sequences

To remove a sequence from the unit, first select (highlight) the sequence (or list of sequences) to be removed, then click on the **REMOVE SEQUENCE** button.

## Selecting a Specific Sequence for SFC, Sequence Detail, or Debug Display



To select a specific sequence for SFC, Sequence Detail, or Debug display, select the Sequence ID, right click for the context menu and select the SFC, Sequence Detail Display or Sequence Debug Display.

## Sequence Detail Display

The Sequence Detail Display, [Figure 34](#), supports runtime monitoring and manipulation of an individual sequence. Most of the runtime changes made to a sequence are done from this display. The changes you can make include:

- Step manipulation (up to 255 steps supported)
- Sequence state, mode, or status

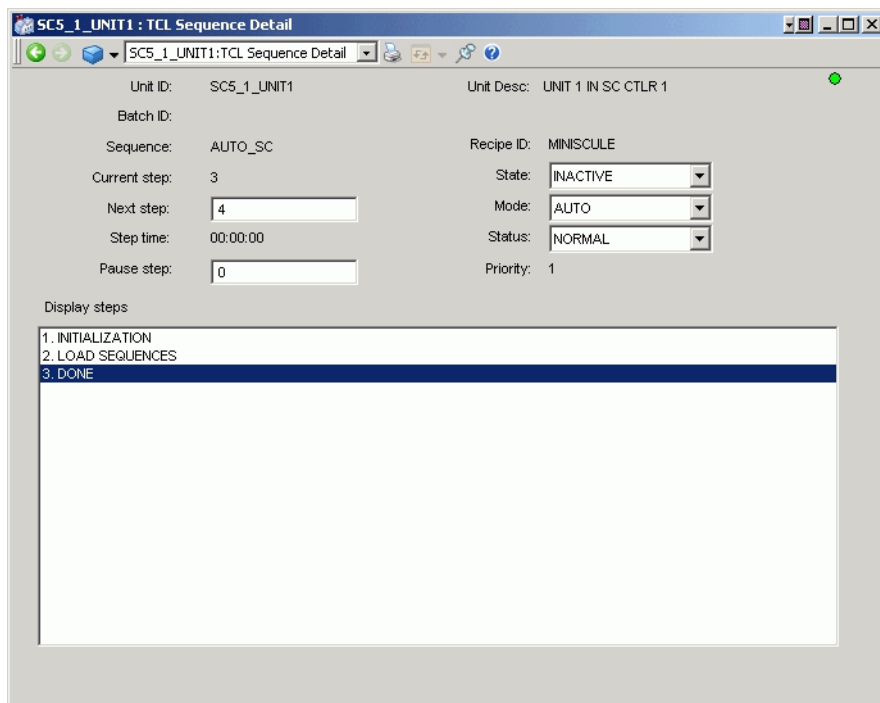


Figure 34. Sequence Detail Page



A limit of 255 steps can be displayed. Additional steps are indicated by the text: “\*TEXT NOT AVAILABLE\*”. These steps can be expanded on the SFC display to get their data.

## Sequence Detail Display Format

The unit ID, recipe ID, batch ID, and sequence ID are shown at the top of the Sequence Detail Display. Step data fields show the current step, next step, step time (the time spent executing the current step), and pause step (or the step on which the program pauses in semiauto mode). There are also fields that show the current state, mode, priority, and status. The Display Steps field lists the sequence steps in numerical order, with the currently executing step highlighted.

## Choosing a Sequence for Sequence Detail

Direct access of a sequence detail is possible from the MOD\_SEQUENCE object which specifies a valid sequence as defined during configuration. Otherwise, a listing of TCL sequences is provided when you first enter a MOD Sequence Detail display from a MOD\_UNIT object. Choose the sequence to be loaded from the list on the Sequence List display, [Figure 35](#).

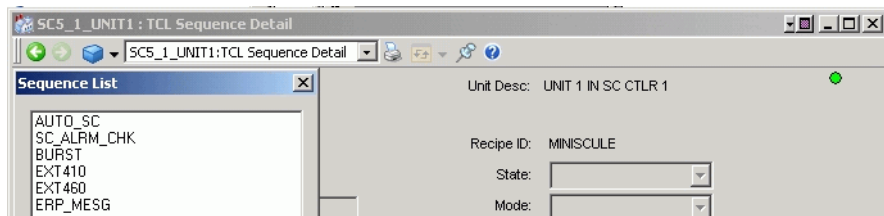


Figure 35. TCL Sequence List

## Changing Sequence State, Mode, and Status

To change state, mode, or status, use the pull-down list and make your selection.

## Manipulating Sequence Steps

Sequence steps can be operated manually one at a time or semi-automatically. Steps can be skipped and they can also be repeated. To make step changes, first select the step, and then enter the new values in the Next Step and Pause Step boxes. Next Step causes the step indicated to be executed after the current step. The next step can skip or repeat other steps as necessary. Pause Step determines at which step the sequence stops and waits for an operator response.



## SFC (Sequential Function Chart) Display

The SFC (Sequential Function Chart) Display, [Figure 36](#), supports runtime monitoring and manipulation of an individual sequence by presenting information, similar to that found in the Sequence Detail Display, but with a graphical flow chart format. This chart presents TCL steps and activities (actions a sequence will execute), along with the transition conditions which determine the flow of the sequence execution. Activities and conditions describe the program flow in plain language rather than the TCL syntax used in the Sequence Debug Display.

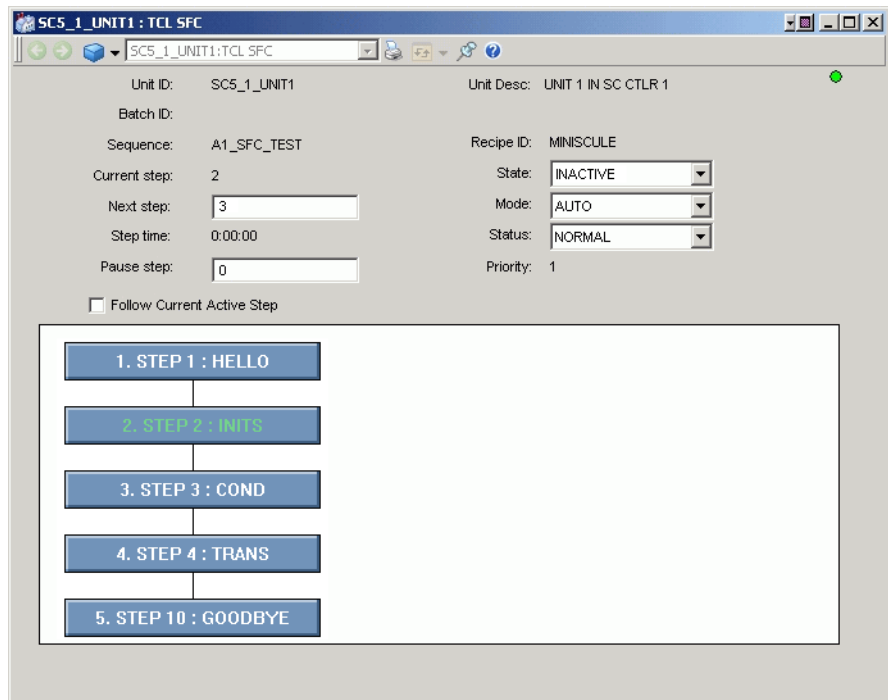


Figure 36. SFC Display

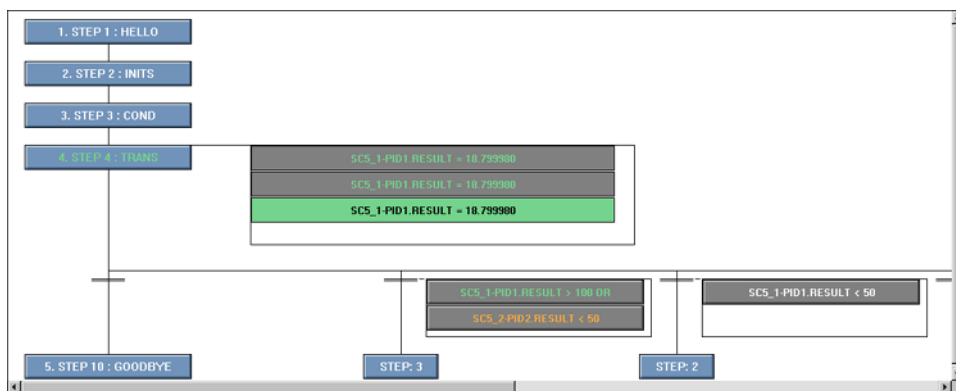
## SFC Display Format

The Unit ID and Description, Recipe ID, Batch ID, and Sequence are shown at the top of the SFC Display. Step data fields show the Current step, Next step, Pause step (the step on which the program pauses), and Step time (the time spent executing the current step). Refer to [Manipulating Sequence Steps](#) on page 88.

State, Mode, and Status are changed using a pull-down list to make your selection. For information on using this dialog, refer to [Changing Sequence State, Mode, and Status](#) on page 88.

The Follow Current Active Step check box, when selected, automatically follows the current active step as the sequence progresses. The step that is currently active is shown expanded in the center of the display. You can also expand or collapse a step by clicking on it. Only one step can be expanded at a time.

An expanded step, [Figure 37](#), shows its activities, conditions, and transitions. When a step is expanded, the area to the right of the step displays the activities and conditions associated with the step in an independent window. Activity statements appear flat, and condition statements appear raised. Use the scroll bars to see any activities or conditions that do not fit in the window.



*Figure 37. SFC Display with Expanded Step*

Transitions are shown beneath the activities and conditions window for a step. There are two types of transition windows. If a transition contains a GOTO which addresses the next step in the TCL sequence, it is displayed with the step number

and step text beneath the tic mark on the main program flow line. Transitions which refer to other steps or which do not have GOTO actions are displayed in the expanded step area. If a GOTO is present, the step number is displayed on a button following the transition tic mark. Use the horizontal scroll bar to see additional transitions, if necessary.

Activities and conditions can have both load text and runtime text. The load text is displayed when the sequence is first loaded. It is replaced by runtime text as that part of the sequence is executed. Load text describes what should happen during execution of the sequence; runtime text describes what actually happens. Transitions only have load text. Each transition has two load texts (one for each button).

## SFC Display Operation

### Activities

Step activities specify the actions that will be performed in each program step. Activities are highlighted to illustrate the degree of execution of the program (the point the program has progressed to). There are three highlight states: Base, Last-Executed, and Executed. [Table 6](#) lists the highlight states and their colors.

.

*Table 6. SFC Highlight States*

Presentation	ACTIVITY	CONDITION	TRANSITION
White on gray or blue	Base	Base	Base
Green on gray	Executed	Executed-True	Evaluated-True
Goldenrod on gray	<Not Used>	Executed-False	Evaluated-False
Black on green	Last-Executed	<Not Used>	<Not Used>

### Conditions

Step conditions contain tag references for operator control. There are three highlight states for condition statements: Base, Executed-True (the expression in the statement directly following the condition is true), and Executed-False (the expression is false). Refer to [Table 6](#) for highlight colors.

Selecting a condition provides direct control access to the database reference associated with the condition. You can change the value of the referenced database point in order to satisfy the expression in the statement following the condition.

To gain control access, right click on a transition button, then select either the Loop FCM Display or the faceplate for the loop associated with the transition. You can change loop values from either the Loop FCM Display or faceplate.



Operator access to the database reference is only available after the statement has been executed.

### Transitions

Step transitions contain tag references for operator control. There are three highlight states for transition statements: Base, Evaluated-True (the transition expression is evaluated as true), and Evaluated-False (the transition expression is evaluated as false). Refer to [Table 6](#) for highlight colors.

Selecting a transition provides direct control access to the database reference(s) associated with the transition. You can change the value of the referenced database point to satisfy the transition expression.

To gain control access, right click on a transition button, then select either the Loop FCM Display or the control dialog for the loop associated with the transition. You can change loop values from either the Loop FCM Display or control dialog.



Operator access to the database reference is only available after the statement has been executed.

## Sequence Debug Display

The Sequence Debug Display, [Figure 38](#), supports runtime troubleshooting and debugging of sequences. The troubleshooting and debug functions on this display are trace and breakpoint. Control functions supported by the sequence debug display are: sequence state, mode, and status changes.

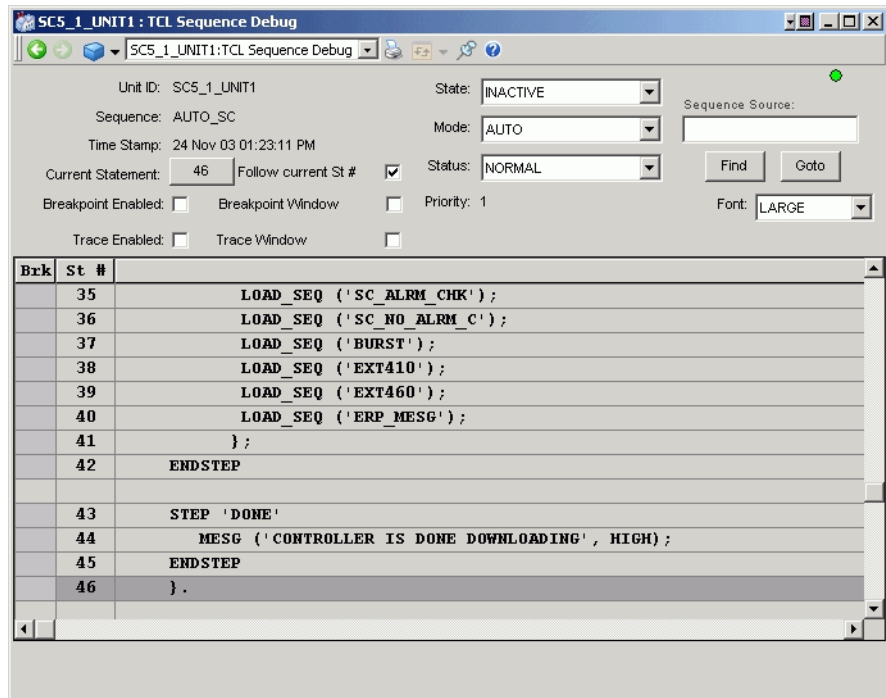


Figure 38. Sequence Debug Display

### Sequence Debug Display Format

The unit ID, sequence ID, time stamp, current statement number, and the sequence state, mode, priority level and status are shown at the top of the display. Below this are toggle buttons that show the status of the [Breakpoint](#) and [Trace](#) debug functions. You can move the contents of the display to a requested point in the program through the [Find](#) and [Goto](#) buttons. A font selection allows you to display large,

medium or small fonts and consequently fewer or more lines in the main portion of the display. The main portion of the display shows a listing of the sequence source statements as they were written using the TCL Editor, and an executable line number for each statement. Note that the [Sequence Source](#) field will appear if an alias is used. At the end of the listing, TCL Compiler Summary and TCL Linker Summary information is included.

## Choosing a Sequence for Debug Display

A listing of TCL sequences is provided when you first enter a MOD Sequence Debug display. Choose the sequence to be loaded from the list on the Sequence List display, [Figure 35](#) (typical).

## Sequence Source

The message “Unable to read sequence source.” is displayed when opening the Sequence Debug display if the source is down, such as when AdvaBuild is not on-line, or when batch uses an alias for the sequence. If an alias is being used, type the sequence name in the Sequence Source box, [Figure 38](#), and press Enter to view the TCL program. If necessary, bring the sequence source on-line and then enter the sequence name.

An alias is automatically created when a sequence is used twice on the same unit. The Sequence Source box allows you to resolve which sequence is to be used. Used within batch, aliases allow an additional copy of a TCL program to be loaded to a unit without having to duplicate the program in the catalog.

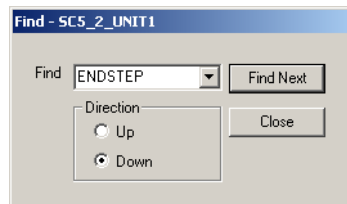
## Changing Sequence State, Mode, and Status

Select state, mode or status conditions from the pull-down lists to make changes.

## Sequence Debug Program Functions

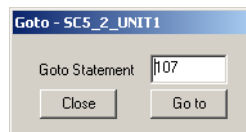
Sequence debug program functions include trace and breakpoint definition. In addition, you can use Find and Goto to move the contents of the display to a requested point in the program. The current statement is highlighted while the program is executing. Select **Follow current St#** to follow the current statement.

## Find



To go to a specific text string in the program, select the **Find** button. This displays the Find window. Select the **Find** field and enter the text string you wish to find (a pull down list lets you select previous text strings). Use **Up** to search backward through the program, or **Down** to search forward through the program. Use **Find Next** to continue the search.

## Goto



To go to a specific line number, select the **Goto** button. This displays the Goto window. Select the **Goto Statement** field and enter the line number you wish to go to. Then click on the **Goto** button. The line you requested is displayed in the source statements window.

## Trace

The trace function tracks the chronological order of the last 12 executed statements during MANUAL program execution. It can be enabled or disabled at any time and may also be applied to up to four variables.

To enable or disable the trace list, click on the **Trace Enabled** check box.

To view the Trace Window, [Figure 39](#), click on the **Trace Window** check box. window shows a Trace Steps list of the last 12 executable statements. You can add up to four trace local or database variables which are of type integer or real in the Trace Window.

To add a trace variable, enter the variable in the text entry field, then press Enter. The variable is then displayed in the list of Trace Variables.

To modify a variable, select the variable in the Trace Variables list and Enter the new value in the text entry field. To delete a variable, select the variable in the Trace Variables list and then click on the **Delete** button. To delete all variables, click on the **Delete All** button.

Click on the **Close** button to close the Trace Window.

Click on the **Cancel** button to deselect a chosen variable, and click on the **Close** button to close the Trace Variable Window.

Trace variables can be created or modified while a sequence is in AUTO mode. This creates a trace variable record in the controller; however, the trace variable record is not applied to the sequence at this time. Trace variable changes are applied to a sequence only when the sequence is in MANUAL mode and tracing is enabled. Any latent (not yet applied) trace variable records are applied when a sequence transitions from AUTO to MANUAL mode (with tracing enabled). Trace variable records can be displayed from any Trace window. For example, you could go over to a different Sequence Debug and see the trace variable not yet applied.

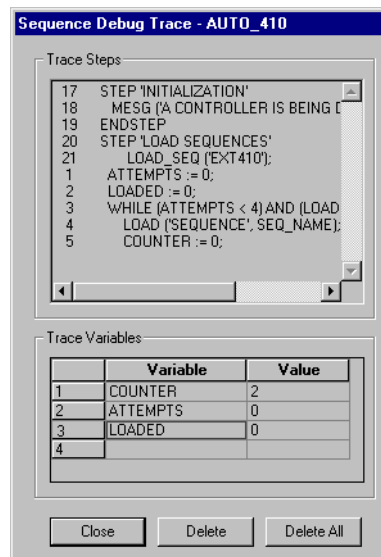



Figure 39. Trace Steps and Variable Window

### Breakpoint

Breakpoints support testing of program logic via manual program execution. When breakpoints are enabled, the sequence is executed until the sequence reaches a line at which a breakpoint has been set. The sequence then enters the paused state, where it remains until either resumed or aborted. While the sequence is in the paused state, tests can be performed and commands can be given to check the sequence logic. The




breakpoint function is enabled and disabled without disturbing breakpoints that have been set.

Breakpoints are indicated on the Sequence Debug Display by a  (green dot symbol) to the left of the step number.

To enable breakpoints, select the **Breakpoint Enabled** box. A breakpoint can be set at five different lines, causing the sequence to execute until it reaches a line at which a breakpoint has been set.

To set a breakpoint at a certain line, either select the **Breakpoint Window** box (this displays the Breakpoint Window, [Figure 40](#)) or click in the **Brk** column (this sets a breakpoint as indicated by a red then green dot). To set the first breakpoint in the **Breakpoint Window**, enter the line number of the line where the first breakpoint is to occur in the **Statement** field. You can repeat this for up to four more breakpoints.

To delete a breakpoint, either select a breakpoint on the Breakpoints list in the **Breakpoint Window** and then click on the **Delete** button, or click on a  (green dot symbol) on the Sequence Debug Display (the dot will change to red then disappear when the breakpoint is deleted).

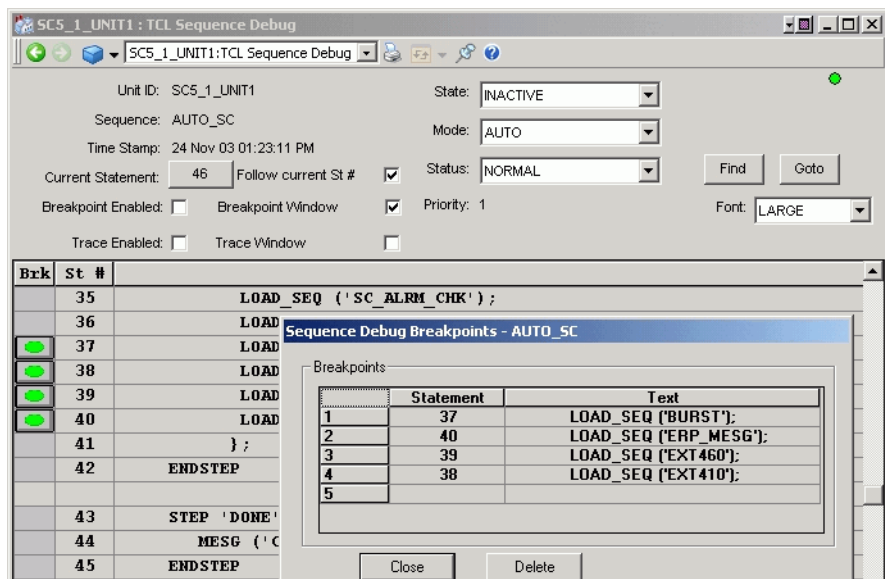


Figure 40. Breakpoint Window

## Recipe Detail

The Recipe Detail Display, [Figure 41](#), shows the recipe items and accompanying data values for the recipe residing on the unit. Runtime changes to recipe parameters are made via this display.

	Item ID:	Description	Value	High Limit	Low Limit	Eng Units
1	INT000	ITEM 0	76	2001	0	INTS
2	REAL104	ITEM 4	10.0	2000.0	0.0	REALS
3	STRITEM	A STRING TYPE	THIS IS A RECIPE STRIng type item value			

Figure 41. Recipe Detail Display

### Recipe Detail Display Format

The unit ID, unit description, recipe ID, and batch ID are shown at the top of the display. The main portion of the display shows the values for each recipe parameter (referred to as recipe item). The display provides the following for each recipe item: ID, description, current value, high and low limits, and engineering units.

## Changing Recipe Values

Only recipe values can be changed from the Recipe Detail Display. To edit a recipe value, select the parameter you want to change, type a new value and press Enter.



Any changes to recipe values from the Recipe Detail Display are not permanent. The changes are temporary and only apply to the current batch.

## TCL Array Plot Display

There are two primary ways plot display are configured: the Plot and the Filled Plot. The Plot display shows data contained in a TCL array as a contiguous trace. As many as two traces of data can be plotted at one time. The Filled Plot configuration, [Figure 42](#), is a single trace, filled with a color between the x-axis and the trace line.

### TCL Array Plot Format

Plot displays require a TCL floating point array to store the configuration attributes and plot point values. The array must be large enough to store values for all the points to be plotted.

Plots and filled plots can have vertical blue lines drawn at user-specified locations. Plots of this type are called Hybrid Plots. Hybrid plot displays are useful for flat sheet applications. A TCL hybrid line array is required for the hybrid plot display. The hybrid line array is a TCL real number array that is 260 elements long.

### Using TCL Array Plot

When using the **In Plot Array** option, the first 10 elements of the array are used as the plot display's attribute values. All other elements of the array may contain the plot point values. The points are drawn on the array plot display consecutively from left to right. Therefore, it is important that you enter the values for the plot points in the proper elements of the array.



The plot array may give you an inaccessible error on first use. Also, anything that causes a disconnect with the Connectivity Server will cause error values. Upon reconnect, inaccessible errors will occur.

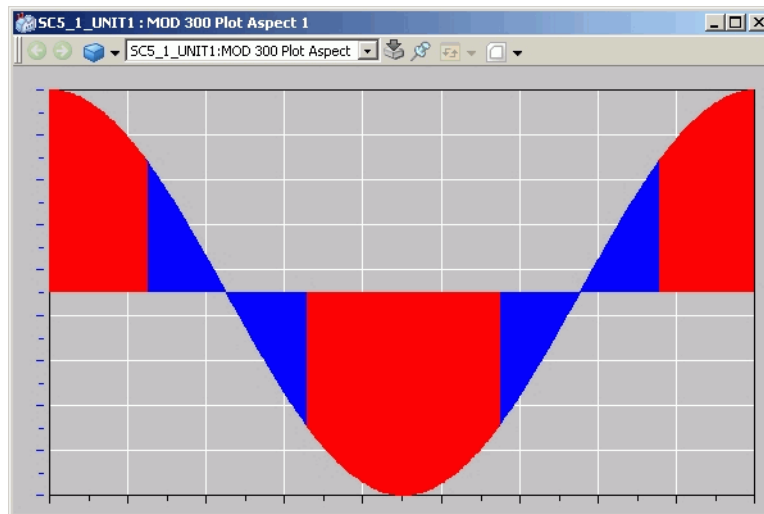


Figure 42. TCL Array Plot

### Context Menu

Use the context menu to perform the following activities:

- **Trace 1 and Trace 2 > Configure**  
This menu is active when using the **In Plot Array** option during configuration. The fields can be set as follows:

<b>Number of Points</b>	This field is used to specify the number of points to be plotted. Number of Points + X-Start Position = X-Axis Width Specifying more points than available will cause a diagnostic error message and will not display any points. Specifying less than 25 points will produce a low resolution plot (the ruler will offset to the nearest point during selection).
-------------------------	--

<b>Y-Axis Top and Y-Axis Top Bottom</b>	This field is used to specify the top and bottom values of the y-axis.
<b>X-Axis Width</b>	This field is used to specify the width of the x-axis for the trace in points. X-Axis Width = Number of Points + X-Start Position For plot displays, traces #1 and #2 can have different x-axis widths. Both x-axes will always be spread out between the y-axes.
<b>X-Start Pos</b>	This field is used to specify the point on the x-axis to start plotting the data pointed to by the respective trace plot array. X-Axis Width = Number of Points + X-Start Position

- **Trace 1 and Trace 2 > Expand to Half Width**  
Use this to view one half of the trace.
- **Trace 1 and Trace 2 > Expand to Quarter Width**  
Use this to view one fourth of the trace.
- **Trace 1 and Trace 2 > Restore**  
Use this to set the default configuration values. There are separate defaults for the Plot and Filled Plot displays.
- **Ruler**  
Use this to show the vertical ruler (checked) or not show the ruler.

**Display Errors**

Use this to view Plot Errors that are the result of configuration errors or an array not being loaded.



---

## Section 5 TLL Displays

### Operational Displays for Taylor Ladder Logic

This section describes the runtime support displays for Taylor Ladder Logic (TLL). These displays are used to monitor TLL segments and data structures. Each data structure has its own type of display (Counter, Register, Timer, IOPoint, File and Sequencer).

The Segment, Counter, Register, Timer and IOPoint displays all include a search feature. To use this feature, type in the search string (can be any string segment) and then select the search arrows to search up (left) or down (right) from the current location. The current match will be colored green (blue on Segment display).

### Segment Display

The Segment Display, [Figure 43](#), is used to monitor and control the execution of Taylor Ladder Logic (see the *TLL User's Guide*). This display is accessed by selecting a Ladder Logic object and then selecting **Segment Display** from the context menu.

The Segment Display can be used to:

- Load and remove segments
- Turn TLL scanning on and off
- Debug segments by forcing the I/O points to specified conditions
- Access displays for the TLL Data Structures (timers, counters, and so on)

The body of the display is a ladder logic diagram. In the left margin, the segment and rung numbers of each ladder rung are indicated. Line numbers are assigned according to the line number used in AdvaBuild. Some line numbers are without a rung because block instructions formerly used two lines for display purposes.

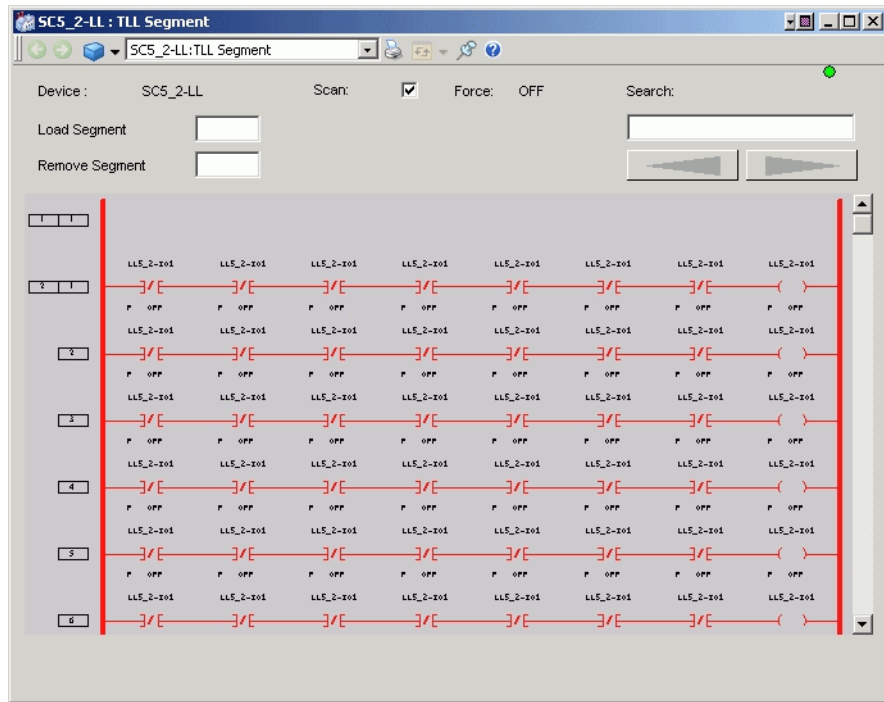


Figure 43. Segment Display

The power rails are displayed in red because they are always powered. The rungs and branches are displayed in red if they have power, or in white if they do not. The elements are labeled with the name that was entered for them when the segment was built. Examine On Symbols show either the state of the contact or the data quality of the input. Timers and counters show either their enable/disable condition or whether they are still running. Remarks are displayed on their own rung. A closed segment also closes its data access subscription (does not collect data in background when closed).

File instructions, [Figure 44](#), allow the ladder logic to transfer data from registers to files, files to registers, and from one file to another. Each instruction has a counter associated with it to provide an index into a file for a particular element (counter faceplates must be accessed from the counter display). The counters are



incremented or decremented by the Ladder Logic to provide the proper index for the Register to File Move and File to Register Move Instructions. The entire contents of a file is moved by the File to File Move Instruction. Not done is indicated by ND on the File to File symbol.

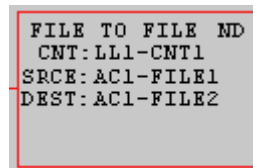


Figure 44. TLL Segment Display File to File Symbol

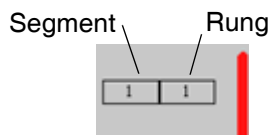
## Segment Display Fields and Buttons

<b>Device</b>	The Device field indicates the TLL name in a specific node or controller. This name is defined on the Ladder Logic Device object during database configuration.
<b>Scan</b>	This selection box is used to toggle the scan on and off. To toggle, click on the <b>Scan</b> box. The scan is on when the box is checked.
<b>Force I/O</b>	The Force I/O field is a display-only field. It indicates whether any I/O point in the segments is being forced by operator command (as read from the database).
<b>Load Segment #</b>	<p>This field is used to add segments to the node or controller. (The segments must already exist in a loadable form for the node or controller.)</p> <p>To add a segment, select the Load Segment # field and enter the number of the segment. Press <b>&lt;Return&gt;</b>.</p> <p>The information on the screen does not change after the load unless the elements on the screen are associated with the newly loaded segment. Otherwise, you have to use the scroll bar to view the newly loaded segment.</p>

**Remove Segment #** This field removes segments from the node or controller. To remove a segment, select the Remove Segment # field and enter the number of the segment. Press **<Return>**.

**Search: (for element)** Use this field to specify an element you wish to find within the Segment display starting with the currently open segment or the first segment if none are open. To find an element, select the search field and enter the name of the element you want to find. If you do not know the full name, enter as much of the name as you know. Click on the right arrow button to search forward or on the left arrow button to search backward. The item you put into the Search field stays there until you replace it with a new item. Changing the search item causes the search to start at the beginning of the segment, if open. If an item is found in a closed segment, you are asked if you want to search in that segment. Selecting No ends the search. The search does not wrap back to the first segment. To start from the beginning, open the first segment and start the search.

### Segment Buttons



Use the segment box to switch between the compressed and expanded segment view. In compressed view, no rungs are displayed and no data access for that segment occurs. In expanded view, all rungs for the segment are visible, and all data access to support the segment is enabled.

By default, all segments are closed. Only one segment can be open at a time. The ability to open all segments is not implemented to prevent communication overload.

## Faceplates for Program Elements

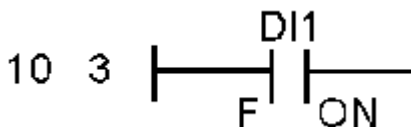
Faceplates provide detailed information about selected ladder elements. To call up a faceplate, click on a program element. Counter faceplates must be accessed from the counter display and not from file instruction elements.

### TLL I/O Point Faceplate

The TLL I/O Point faceplate allows you to debug segments by specifying states for the AC parameter of I/O points.

To change AC, you must first put the I/O point into a forced state by clicking on the Forced **YES** button. Then you can change the status between **ON** and **OFF** or **OPEN** and **CLOSED**. When a point is forced, it affects the value in the segment only. The actual condition of the point in the field is not affected.

The character **F** is used to indicate all points that are forced, [Figure 45](#). Forcing is also supported via the I/O Point Display. For more information, refer to [I/O Point Display](#) on page 115.



*Figure 45. Indication of a Forced Point*

### TLL Timer Faceplate

You can change the Preset, Value, and Enable in the TLL Timer faceplate. The format for Preset and Value time entries is described in [Timer Display](#) on page 112.

### TLL Counter Faceplate

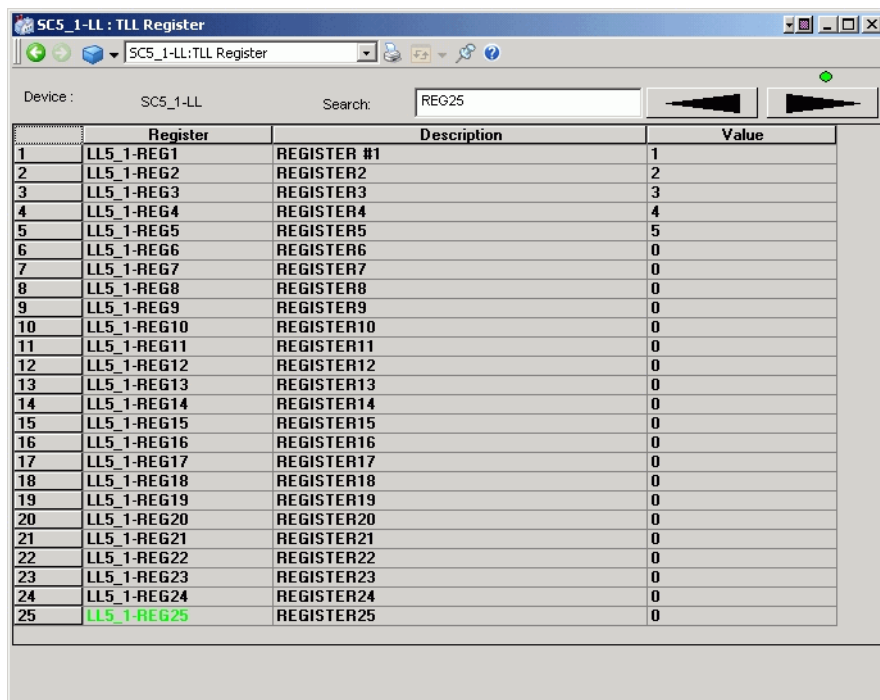
You can change the Preset, Value, and Enable in the TLL Counter faceplate. For more information on the TLL Counter faceplate, refer to [Counter Display](#) on page 110.

### TLL Register Faceplate

You can change the Value in the TLL Register faceplate. For more information on the TLL Register faceplate, refer to [Register Display](#) on page 108.

## Register Display

The Register Display, [Figure 46](#), contains a list of the registers, places in memory that are used to store values, in the device. This display is accessed by selecting a Ladder Logic object and then selecting **Register Display** from the context menu.



The screenshot shows a software window titled "SC5\_1-LL : TLL Register". The window contains a table with the following data:

	Register	Description	Value
1	LL5_1-REG1	REGISTER #1	1
2	LL5_1-REG2	REGISTER2	2
3	LL5_1-REG3	REGISTER3	3
4	LL5_1-REG4	REGISTER4	4
5	LL5_1-REG5	REGISTER5	5
6	LL5_1-REG6	REGISTER6	0
7	LL5_1-REG7	REGISTER7	0
8	LL5_1-REG8	REGISTER8	0
9	LL5_1-REG9	REGISTER9	0
10	LL5_1-REG10	REGISTER10	0
11	LL5_1-REG11	REGISTER11	0
12	LL5_1-REG12	REGISTER12	0
13	LL5_1-REG13	REGISTER13	0
14	LL5_1-REG14	REGISTER14	0
15	LL5_1-REG15	REGISTER15	0
16	LL5_1-REG16	REGISTER16	0
17	LL5_1-REG17	REGISTER17	0
18	LL5_1-REG18	REGISTER18	0
19	LL5_1-REG19	REGISTER19	0
20	LL5_1-REG20	REGISTER20	0
21	LL5_1-REG21	REGISTER21	0
22	LL5_1-REG22	REGISTER22	0
23	LL5_1-REG23	REGISTER23	0
24	LL5_1-REG24	REGISTER24	0
25	LL5_1-REG25	REGISTER25	0

Figure 46. TLL Register Display

The following fields can be edited on this display:

- Value                    You can change the value of a register by selecting the Value field for the register, entering the new value, then pressing **ENTER**. This change is then applied.
- Description            Use this field to modify the associated description.

Search is case sensitive and will find any partial string in the Register column. A match is colored green. The search arrows will find the previous or next match.

## Register Faceplate

Each register can have a faceplate. Select the register and use the context menu to select the faceplate as show in [Figure 47](#). You can change the Description and value from the faceplate.

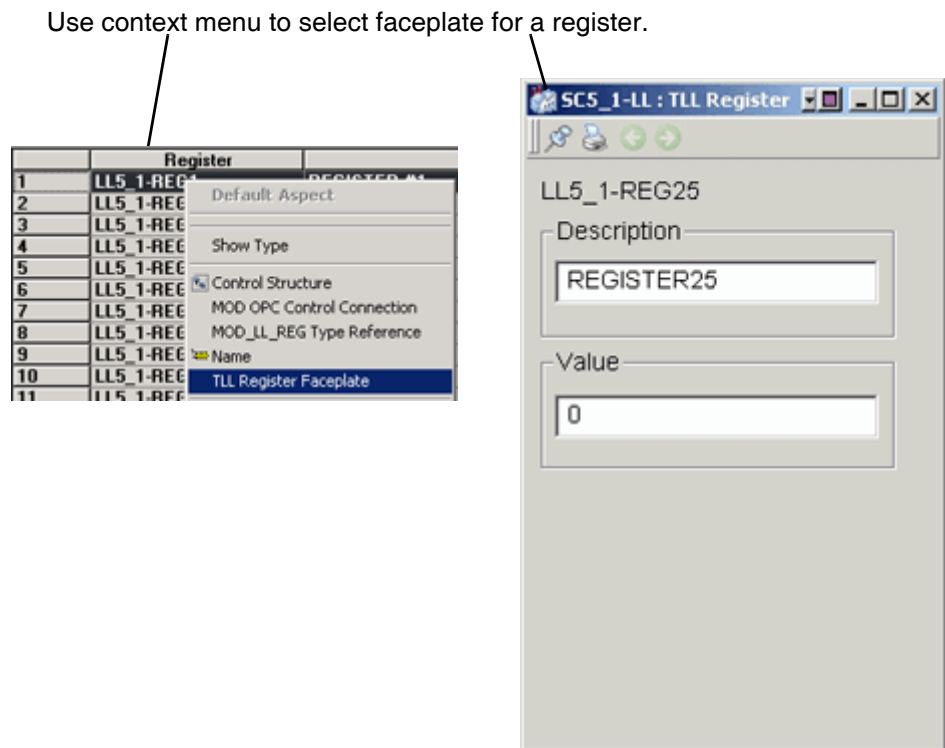


Figure 47. TLL Register Faceplate

## Counter Display

The Counter Display, [Figure 48](#), contains a list of the counters in the device. This display is accessed by selecting a Ladder Logic object and then selecting **Counter Display** from the context menu.

TLL counters can count by increments of one. Instructions in the programs can cause them to count up or down, reset, or go to their configured preset values. Counters can count as high as 2,147,483,647. Counters are required by a number of instructions such as file instructions and sequencer instructions. Down counting starts at the Preset and is finished when the count Value = 0. Up counting starts at the count Value = 0 and is finished when equal to the Preset.

	Counter	Description	Preset	Value
1	LL5_1-CNT1	COUNTER #1	1234567890	0
2	LL5_1-CNT2	COUNTER2	1111111111	0
3	LL5_1-CNT3	COUNTER3	0	0
4	LL5_1-CNT4	COUNTER4	0	0
5	LL5_1-CNT5	COUNTER5	0	0
6	LL5_1-CNT6	COUNTER6	0	0
7	LL5_1-CNT7	COUNTER7	0	0
8	LL5_1-CNT8	COUNTER8	0	0
9	LL5_1-CNT9	COUNTER9	0	0
10	LL5_1-CNT10	COUNTER10	0	0
11	LL5_1-CNT11	COUNTER11	0	0
12	LL5_1-CNT12	COUNTER12	0	0
13	LL5_1-CNT13	COUNTER13	0	0
14	LL5_1-CNT14	COUNTER14	0	0
15	LL5_1-CNT15	COUNTER15	0	0
16	LL5_1-CNT16	COUNTER16	0	0
17	LL5_1-CNT17	COUNTER17	0	0
18	LL5_1-CNT18	COUNTER18	0	0
19	LL5_1-CNT19	COUNTER19	0	0
20	LL5_1-CNT20	COUNTER20	0	0
21	LL5_1-CNT21	COUNTER21	0	0
22	LL5_1-CNT22	COUNTER22	0	0
23	LL5_1-CNT23	COUNTER23	0	0
24	LL5_1-CNT24	COUNTER24 asdf	1	1
25	LL5_1-CNT25	COUNTER25	0	0
26	LL5_1-CNT26	JOSH TEST	0	0

Figure 48. TLL Counter Display

The following fields can be edited on this display:

Value	You can change the value of a counter by selecting the Value field for the counter, entering the new value, then pressing <b>ENTER</b> . This change is logged.
Preset	You can change the preset of a counter (the value counted up to or down from) by selecting the Preset field for the counter, entering the new value, then pressing <b>ENTER</b> . This change is logged.
Description	Use this field to modify the associated description.

## Counter Faceplate

Each counter can have a faceplate, [Figure 49](#). Select the counter and use the context menu to select the faceplate. The counter can be enabled (YES) or disabled (NO) from the faceplate. You can also change the Description, Preset and Value.

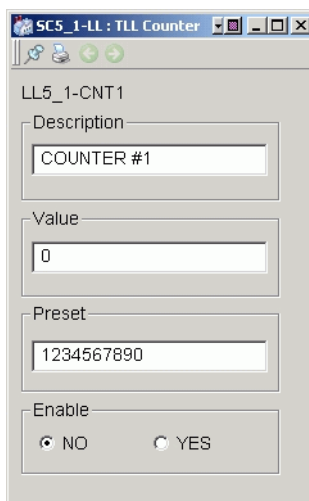


Figure 49. TLL Counter Faceplate

## Timer Display

The Timer Display, [Figure 50](#), contains a list of the timers in the device. This display is accessed by selecting a Ladder Logic object and then selecting **Timer Display** from the context menu.

TLL timers can time with a configurable time base that can be either 1.0, 0.1, or 0.01 seconds (displayed on faceplate). The timers are under program control and can be started, stopped and reset by program instructions. They can time to values as large as 999 hours. A timer is configured to time up or down.

The screenshot shows a software window titled "SC5\_1-LL : TLL Timer". The window contains a table with the following columns: "Timer", "Description", "Preset", "Value", and "Enabled". The table lists 25 timers, each with a unique ID (e.g., LL5\_1-TIM1), a description (e.g., TIMER #1), a preset value (e.g., 000:00:01.00), a current value (e.g., 000:00:00.00), and an enabled status (e.g., NO).

	Timer	Description	Preset	Value	Enabled
1	LL5_1-TIM1	TIMER #1	000:00:01.00	000:00:00.00	NO
2	LL5_1-TIM2	TIMER2	000:00:01.00	000:00:00.00	NO
3	LL5_1-TIM3	TIMER3	000:00:00.00	000:00:00.00	NO
4	LL5_1-TIM4	TIMER4	000:00:00.00	000:00:00.00	NO
5	LL5_1-TIM5	TIMER5	000:00:00.00	000:00:00.00	NO
6	LL5_1-TIM6	TIMER6	000:00:00.00	000:00:00.00	NO
7	LL5_1-TIM7	TIMER7	000:00:00.00	000:00:00.00	NO
8	LL5_1-TIM8	TIMER8	000:00:00.00	000:00:00.00	NO
9	LL5_1-TIM9	TIMER9	000:00:00.00	000:00:00.00	NO
10	LL5_1-TIM10	TIMER10	000:00:00.00	000:00:00.00	NO
11	LL5_1-TIM11	TIMER11	000:00:00.00	000:00:00.00	NO
12	LL5_1-TIM12	TIMER12	000:00:00.00	000:00:00.00	NO
13	LL5_1-TIM13	TIMER13	000:00:00.00	000:00:00.00	NO
14	LL5_1-TIM14	TIMER14	000:00:00.00	000:00:00.00	NO
15	LL5_1-TIM15	TIMER15	000:00:00.00	000:00:00.00	NO
16	LL5_1-TIM16	TIMER16	000:00:00.00	000:00:00.00	NO
17	LL5_1-TIM17	TIMER17	000:00:00.00	000:00:00.00	NO
18	LL5_1-TIM18	TIMER18	000:00:00.00	000:00:00.00	NO
19	LL5_1-TIM19	TIMER19	000:00:00.00	000:00:00.00	NO
20	LL5_1-TIM20	TIMER20	000:00:00.00	000:00:00.00	NO
21	LL5_1-TIM21	TIMER21	000:00:00.00	000:00:00.00	NO
22	LL5_1-TIM22	TIMER22	000:00:00.00	000:00:00.00	NO
23	LL5_1-TIM23	TIMER23	000:00:00.00	000:00:00.00	NO
24	LL5_1-TIM24	TIMER24	000:00:00.00	000:00:00.00	NO
25	LL5_1-TIM25	TIMER25	000:00:00.00	000:00:00.00	NO

Figure 50. TLL Timer Display



The following fields can be edited on this display:

Preset Value	The Preset and Value fields are displayed to the hundredth place. For example, 001:20:15.25 is one hour, twenty minutes, fifteen and twenty-five hundredths seconds. Your entry should be within the resolution of the time base. If it is not, the system automatically rounds it to the nearest multiple of the time base. You can change the preset or value of a timer by selecting the Preset or Value field for the timer, entering the new value, then pressing <b>ENTER</b> . The changes are logged.
Enable	This field is used to enable and disable the timer. You can change between <b>YES</b> (enable) and <b>NO</b> (disable) by selecting the Enable field for the timer, clicking on the appropriate Enable button, then pressing <b>ENTER</b> . The changes are logged.
Description	Use this field to modify the associated description.

Time entries can be made in the following formats:

- One colon preceded by one or two digits means minutes and seconds.

Examples:

30:05	30 minutes 5 seconds
12:40	12 minutes 40 seconds

- One colon preceded by three digits means hours and minutes.

Example:

010:15	10 hours 15 minutes
--------	---------------------

- Two colons indicate hours and minutes.

Examples:

8:07:	8 hours, 7 minutes
12:16:10.2	12 hours, 16 minutes, 10.2 seconds

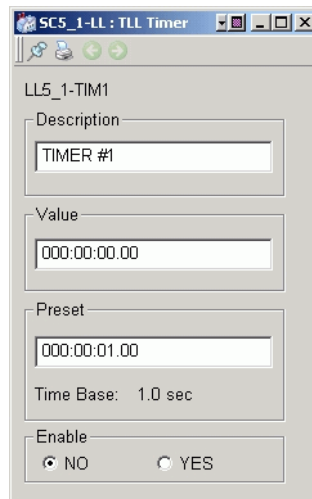
- A period indicates seconds.

Examples:

6.	6 seconds
.5	.5 seconds

## Timer Faceplate

Each timer can have a faceplate, [Figure 51](#). Select the timer and use the context menu to select the faceplate. The timer can be enabled (YES) or disabled (NO) from the faceplate.

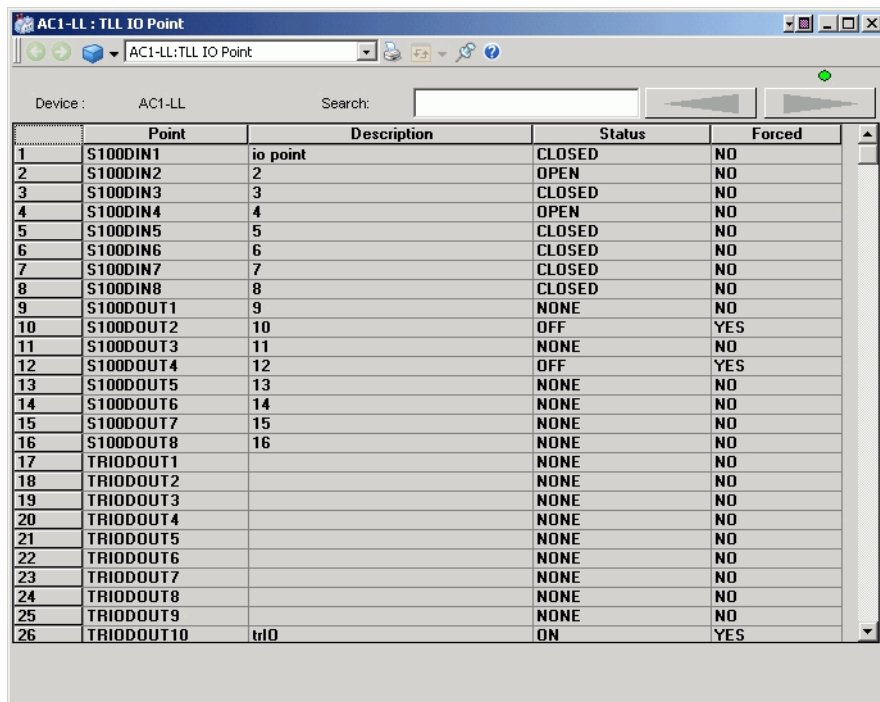


*Figure 51. TLL Timer Faceplate*

## I/O Point Display

The I/O Point Display, [Figure 52](#), contains a list of the I/O points of the node or controller. This display is accessed by selecting a Ladder Logic object and then selecting **I/O Point Display** from the context menu.

TLL uses digital input (contact) and output (coil) points that are local to the controller or node. For ladder logic I/O points that use PLC functionality, only inputs are supported.



The screenshot shows a software window titled "AC1-LL: TLL IO Point". The window contains a table with the following columns: Point, Description, Status, and Forced. The table lists 26 points, including digital inputs (S100DIN1-8), digital outputs (S100DOUT1-8), and tri-state outputs (TRI00DOUT1-10). The status of each point is indicated, such as "CLOSED", "OPEN", "NONE", or "OFF".

	Point	Description	Status	Forced
1	S100DIN1	io point	CLOSED	NO
2	S100DIN2	2	OPEN	NO
3	S100DIN3	3	CLOSED	NO
4	S100DIN4	4	OPEN	NO
5	S100DIN5	5	CLOSED	NO
6	S100DIN6	6	CLOSED	NO
7	S100DIN7	7	CLOSED	NO
8	S100DIN8	8	CLOSED	NO
9	S100DOUT1	9	NONE	NO
10	S100DOUT2	10	OFF	YES
11	S100DOUT3	11	NONE	NO
12	S100DOUT4	12	OFF	YES
13	S100DOUT5	13	NONE	NO
14	S100DOUT6	14	NONE	NO
15	S100DOUT7	15	NONE	NO
16	S100DOUT8	16	NONE	NO
17	TRI00DOUT1		NONE	NO
18	TRI00DOUT2		NONE	NO
19	TRI00DOUT3		NONE	NO
20	TRI00DOUT4		NONE	NO
21	TRI00DOUT5		NONE	NO
22	TRI00DOUT6		NONE	NO
23	TRI00DOUT7		NONE	NO
24	TRI00DOUT8		NONE	NO
25	TRI00DOUT9		NONE	NO
26	TRI00DOUT10	trIO	ON	YES

Figure 52. TLL I/O Point Display

## I/O Point Faceplate

Each I/O point can have a faceplate, [Figure 53](#). Select the I/O point and use the context menu to select the faceplate. For S100 I/O points, the faceplate shows the

Channel, Lan and Block number. For S800 I/O points, the faceplate shows the Channel number. The following fields can be edited on this display:

- |             |  |
|-------------|--|
| Status      | The Status field gives the current value of the AC parameter of the point. It can be either Open or Closed for a contact, or On or Off for a coil. Change the status by selecting the Status field for the point, then selecting the appropriate status from the pull down list. Status changes are logged.  |
| Forced      | When the Forced field is YES, you can determine how the value of a points status affects the segment since the change is not sent to the actual I/O point. When the Forced field is NO, status field changes affects the actual I/O point. You can change between YES and NO by selecting the Forced field for the point, then selecting the appropriate Forced value from the pull down list. Changes to this field are not logged. |
| Description | Use this field to modify the associated description.   |

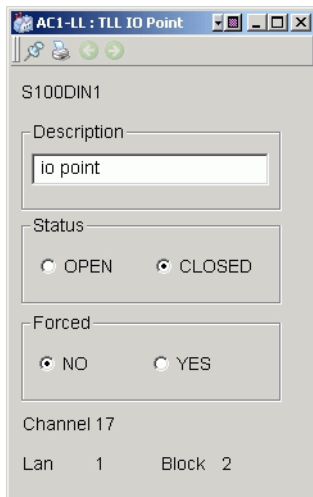
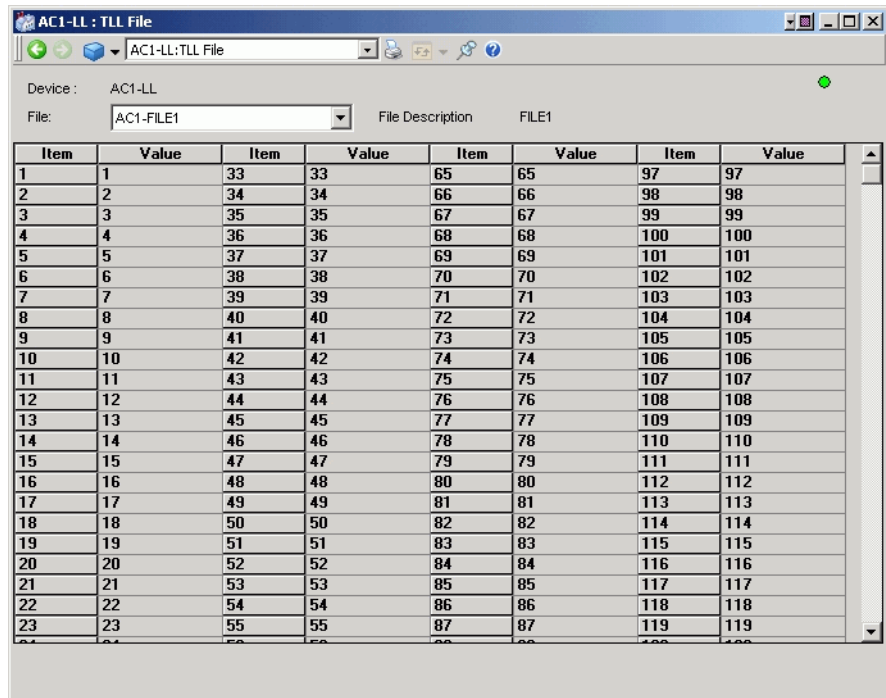


Figure 53. TLL I/O Point Faceplate

## File Display

Each file in the device has a File Display which lists up to 128 values for use by the sequencer, [Figure 54](#). This display is accessed by selecting a Ladder Logic Device object and then selecting **File Display** from the context menu and then selecting a specific file from the File List.



The screenshot shows a window titled "AC1-LL : TLL File". The "Device" is set to "AC1-LL" and the "File" is "AC1-FILE1". The main area contains a table with 128 rows and 8 columns. The columns are labeled "Item" and "Value" in pairs. The values in the "Value" columns are sequential integers from 1 to 128.

Item	Value	Item	Value	Item	Value	Item	Value
1	1	33	33	65	65	97	97
2	2	34	34	66	66	98	98
3	3	35	35	67	67	99	99
4	4	36	36	68	68	100	100
5	5	37	37	69	69	101	101
6	6	38	38	70	70	102	102
7	7	39	39	71	71	103	103
8	8	40	40	72	72	104	104
9	9	41	41	73	73	105	105
10	10	42	42	74	74	106	106
11	11	43	43	75	75	107	107
12	12	44	44	76	76	108	108
13	13	45	45	77	77	109	109
14	14	46	46	78	78	110	110
15	15	47	47	79	79	111	111
16	16	48	48	80	80	112	112
17	17	49	49	81	81	113	113
18	18	50	50	82	82	114	114
19	19	51	51	83	83	115	115
20	20	52	52	84	84	116	116
21	21	53	53	85	85	117	117
22	22	54	54	86	86	118	118
23	23	55	55	87	87	119	119

Figure 54. File Display

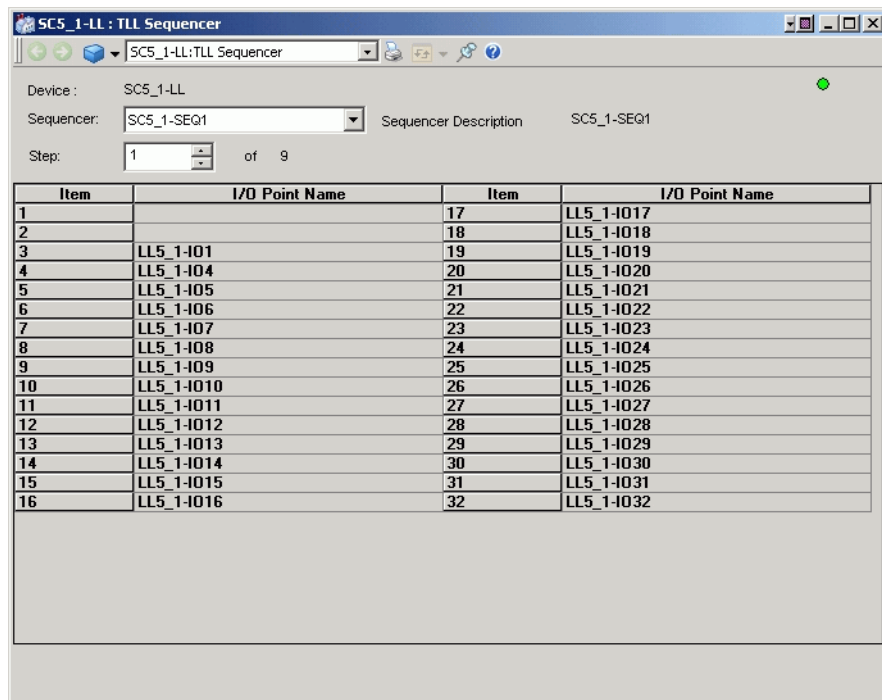
Use the drop down file list to switch to a display for another file in the device. Select the Value field to change the values of the items in the file. All changes are logged.

When changing a value, type in the new value and press **ENTER**. The maximum value allowed is an eight digit integer. For control purposes, 16777215 (FFFFFF hex) is the full access upper value. During TLL execution, the sequencer instructions translate the entry into a bit pattern to load into or compare with I/O points. It is possible to see values beyond what you can enter here because the

LL\_FILE template allows a base 10 integer, from 0 to 4,294,967,295 (32 bits); however, it will be displayed as a signed 32 bit number (-1).

## Sequencer Display

The Sequencer Display, [Figure 55](#), has multiple pages, with one page for each step in the sequencer. This display is accessed by selecting a Ladder Logic Device object and then selecting **Sequencer Display** from the context menu and then a specific sequence from the Sequence List.



Item	I/O Point Name	Item	I/O Point Name
1		17	LL5_1-I017
2		18	LL5_1-I018
3	LL5_1-I01	19	LL5_1-I019
4	LL5_1-I04	20	LL5_1-I020
5	LL5_1-I05	21	LL5_1-I021
6	LL5_1-I06	22	LL5_1-I022
7	LL5_1-I07	23	LL5_1-I023
8	LL5_1-I08	24	LL5_1-I024
9	LL5_1-I09	25	LL5_1-I025
10	LL5_1-I010	26	LL5_1-I026
11	LL5_1-I011	27	LL5_1-I027
12	LL5_1-I012	28	LL5_1-I028
13	LL5_1-I013	29	LL5_1-I029
14	LL5_1-I014	30	LL5_1-I030
15	LL5_1-I015	31	LL5_1-I031
16	LL5_1-I016	32	LL5_1-I032

*Figure 55. Sequencer Display*

A sequencer is a series of up to 128 steps. Each step is a list of as many as 32 I/O points. The I/O points listed in a sequencer step are manipulated in one way or another by sequencer program instructions. One type of sequencer instruction

causes the values of input points to be read into a file. Another instruction causes the values in a file to be sent to output points.

Use the **Step** field to display the screen for a particular step of the sequencer. Use the up/down arrow of the **Step** field to display the screen for the previous/next step of the sequencer. Enter a step value to go to a particular step.

You can switch to a display for another sequencer in the device by selecting one from the **Sequencer** field.

To change a value, select the value field for the Item to be changed, type in a valid I/O point (no spaces), and then press **ENTER**.

## TLL Device Logged Changes

There are several changes that can be made to items on the Database Summary Displays. Almost all of these changes are logged. For example, if your system is configured with a logging printer, a typical message would be:

```
REG1      SEQ1 MASK REGISTERAC CHANGED FROM 115 TO 315
```

## TLL Messages

TLL messages are treated like TCL messages and can be displayed on the TCL Message Display for a unit. TLL messages are configured on the Ladder Logic Message Templet as described in the TLL User's Guide. Units receiving TLL messages during runtime are specified on the TLL Configuration Area Templets.





---

## Section 6 Status Displays

### Status Displays Overview

Status displays contain system and subsystem level information that allow you to identify detectable fault conditions, monitor general status and performance, change process outputs directly for testing, and control the status of redundancy. The displays include: [System Status](#), Subsystem Status ([AC460](#), [AC410](#), [Controller Node](#), [Turbo Node](#)), [Diagnostic Message](#), and [System Performance](#). See [Section 7, I/O Displays](#) for: [S800](#) and [S100](#) Device Overview and Status, [PROFIBUS](#), [TRIO](#) and [Direct I/O](#). For additional information on diagnostic displays for the Advant Controller subsystems, refer to the *Advant Controller 410 User's Guide* and/or the *Advant Controller 460 User's Guide*. For all other subsystems, refer to the appropriate hardware user's guide book.

You are alerted to the presence of a diagnostic message by the Diagnostic Message List of the base product as described in *Industrial IT 800xA System Operation*.

Entries identifying an operator action on the diagnostic displays are relayed to the appropriate logging device.

Color coding of data shown on the diagnostic displays, including the labels on the message targets/buttons, is as follows:

- Red indicates an error,
- green a good report,
- yellow a possible problem, and
- white a neutral fact.

## System Status Display

The System Status Display, [Figure 56](#), shows the current status of all subsystems (nodes) recognized on the DCN. Each subsystem status icon lists the subsystem name, device address, device type, media state, device state, and controller status (Controller types only). The display has a large icon, small icon and a report view available through the context menu (shown). The Address and Type columns may be sorted by clicking on the column heading. Module message and performance displays can be viewed right clicking in the Status column (shown).

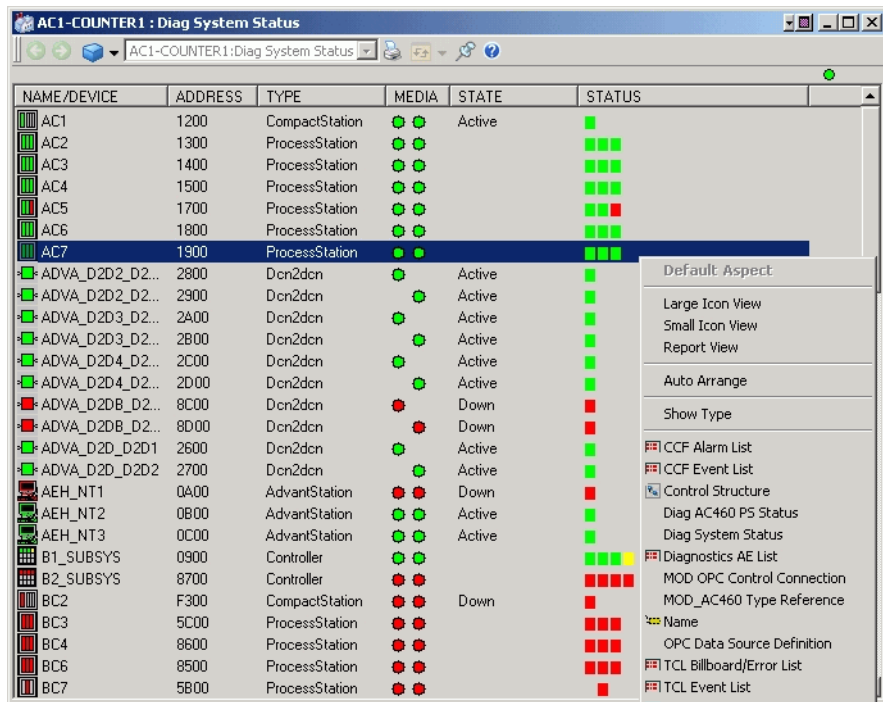


Figure 56. System Status Display, Report View

## Subsystem Types

The available subsystem types (shown above icon on icon views and in the heading line of the corresponding Subsystem or I/O Status Display) include:

ADVANT STATION	Advant Station
COMPACT STATION	Advant Controller 410 (AC410) Subsystem
PROCESS STATION	Advant Controller 460 (AC460) Subsystem
CONSOLE	Console Subsystem
CONTROLLER	Controller Subsystem
DATA PROCESSOR	Turbo Subsystem
DCN2DCN	DCN/DCN Interface
GATEWAY	Ethernet Interface
UNKNOWN	Subsystem not identifiable

## Address

The device number address for an Advant Station, AC410, AC460, Console, Data Processor Subsystem, DCN/DCN Interface, or Gateway Interface is determined by configuration. Subdevice numbers, for the Controller Subsystem only, are determined by the module position in the controller card file and are shown in the Subsystem Status Display.

## Subsystem Name

The subsystem name shown for each icon of the System Status Display is the 21 character name entered in the TEMPLET NAME field when that subsystem was configured.

## Subsystem Status

The status of each subsystem DCN connection is indicated by its color coded status boxes. The number of status boxes depends on the type of subsystem. CONSOLE, DATA PROCESSOR, and GATEWAY icons contain two status boxes that indicate the status of D/M Module 1 and 2. CONTROLLER icons contain two status boxes that indicate the status of D/F Module 1 and 2 and up to 12 status boxes that indicate

the status of Control Modules 1 through 12. ADVANT STATION icons contain two status boxes that indicate the status of the Real Time Accelerator Board. AC410 and AC460 icons contain two status boxes that indicate the status of the DCN interface submodules, and AC460 icons have up to three additional status boxes that indicate the status of controllers.

## State Definitions

The system state executive indicates the current status of a device or subdevice through the following state definitions:

*Table 7. State Definitions*

STATE	COLOR	DEFINITION
	Blank	No Control Module configured.
ACTIVE	Green	Device or subdevice is running.
AVAILABLE	Cyan	Program has downloaded and database is downloading.
DOWN	Red	Fatal failure (that is, communications has failed) or device/subdevice may be down.
HOLD	Cyan	Hardware error (that is, digital input, analog input) or configured hold to allow other Control Module with softwired data to become active.
IMPAIRED	Cyan	Backup Control Module running but is not yet rebooted.
READY	Yellow	Database download is complete; normal state for backup controller.
UNKNOWN	Red	State is not known.

## Subsystem Status Displays

The Subsystem Status Displays provide information on the node's condition.

### AC460 Subsystem Status Display

The AC460 Subsystem Status Display, [Figure 57](#), provides: node address and name; controller address, name and status (active and backup); status of power supplies; TC520 Monitor Status; fan and auxiliary input status; status for PM510 Processor Module Cards and whether a card is currently a primary or backup; status and address for DCN, TRIO, MVI, AF 100 - S800 I/O and S100 I/O submodules; TRIO, S100, S800 and PROFIBUS logical LAN numbers; MVI Port type and status (A/B).

Select a controller to perform Switchover, Hard Reset or Reset for Upgrade activities. Use the context menu to select the Diagnostic Message and System Performance displays for a specific subdevice. Select the [S800](#), [S100](#), [PROFIBUS](#), or [TRIO](#) Overview (LAN) display by selecting the submodule display target.

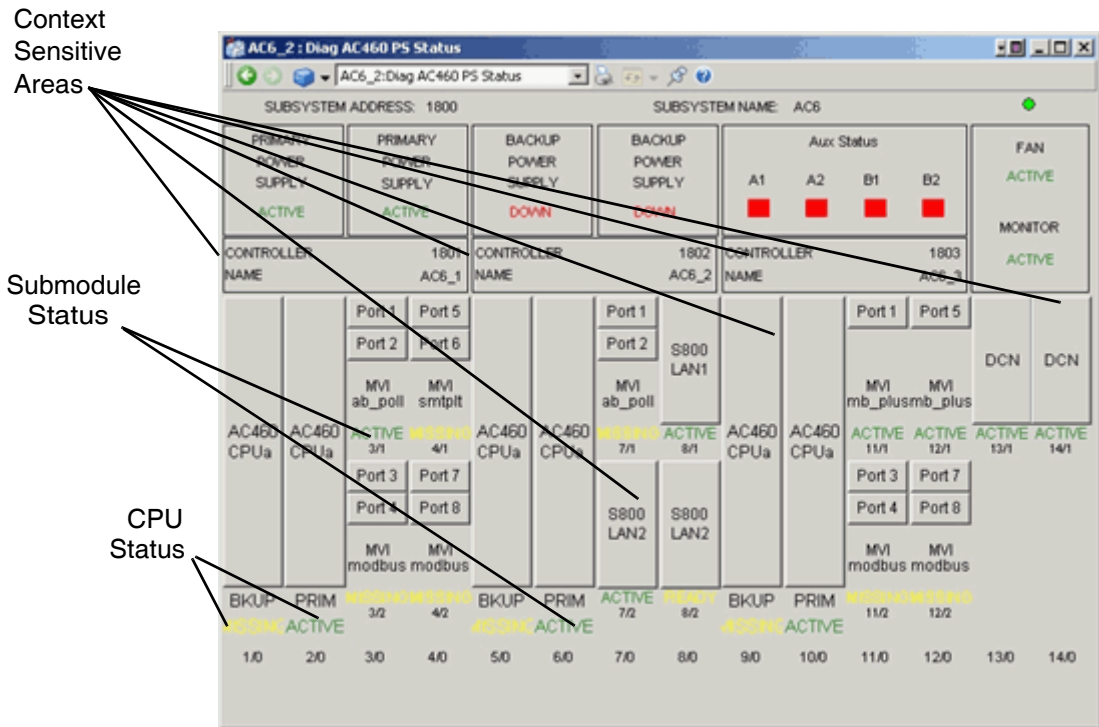


Figure 57. AC460 Subsystem Status Display (Process Station)

**PROFIBUS Interface**

The status of the PROFIBUS Interface will show if it is active, failed or missing. In addition, you can select the PROFIBUS LAN display by selecting its display target.

**TRIO Interface**

The status of the TRIO LAN submodule will show if it is active, failed or missing. Select the TRIO Lan display by selecting its display target.

**S800 I/O Interface**

The status of the S800 I/O Interface submodule will show if it is active, failed or missing. In addition, you can select the [S800 Lan Overview](#) display for a module by selecting its display target.

**MVI Interface**

MVI (Multi Vendor Interface) is redundant at the port level for PLC I/O. MVI redundancy is determined by the controlling application (client) program. Redundant ports may reside on the same, or on a different MVI submodule. Therefore, no MVI submodule is considered to be a “backup” for another. The status of the MVI submodule will show if it is active, failed or missing.

**DCN Interface**

The DCN interface is similar to the TRIO interface, in that both are considered to be active simultaneously, with no preference for either one as primary.

**TC520 Monitor Status**

The TC520 Status Module information includes the power supply status, fan status and user-defined auxiliary status inputs (available from the front panel of the TC520 status monitor). The status can be either DOWN (red) or ACTIVE (green) for all the TC520 functions. If the TC520 monitor status is DOWN, then the status for all the other TC520 functions will be UNKNOWN (yellow). The rectangles for the user-defined auxiliary status inputs (A1, A2, B1, B2) are colored green when active, and colored red when inactive.

### S100 Interface

S100 is redundant at the device level. A device's redundant mate may reside on the same, or on a different S100 LAN. When redundant, both LANs are active, therefore, no S100 LAN interface is considered to be a "backup" for another. The status of the S100 LAN submodule will show if it is active, failed or missing. In addition, you can select the [S100 Overview Display](#) for a module by selecting its display target.

The S100 I/O LAN status show the condition of the CI540 and the LAN communications.

ACTIVE = Primary Redundant LAN is OK

READY = Back-up Redundant LAN is OK

BC DN = Communication to DSBC 175 Bus Extender has a problem

DOWN = CI540 Bus Extender Master has failed

### Push-buttons on AC460 Subsystem Status Display

When a controller is selected, three (3) command push-buttons are displayed, [Figure 58](#). They allow the user to issue commands to the controller in order to perform a software upgrade. When a command push-button is selected, a confirmation is required.

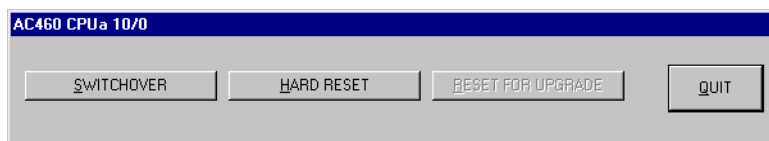


Figure 58. Command Push-buttons



See the *Avant® Controller 460 User's Guide* for a detailed description on how to perform an on-line upgrade.

The descriptions for the three command push-buttons follows:

- **RESET for UPGRADE**

The **RESET for UPGRADE** push-button is used to reset a backup controller and reboot into the “UPGRDE” state. The following restrictions apply for this command:

- The **RESET for UPGRADE** command is used to reset backup controllers only and cannot be invoked from a primary controller. This prevents primary controllers from being reset by the user.
- The command is not actually sent until the **CONFIRM** push-button is selected.
- The **CANCEL** push-button can be used to clear the **RESET for UPGRADE**.

- **SWITCHOVER**

The **SWITCHOVER** push-button is used to force the switchover of a primary controller to a backup controller (that is loaded with a different version of software). The following restrictions apply for this command:

- If the backup controller is not in the “UPGRDE” state, **SWITCHOVER** cannot be selected. This prevents switchover from being attempted for backup controllers that are not in the correct state.
- A five (5) second delay after selection of this push-button occurs before other push-buttons can be selected. This prevents switchover of different controllers in the same AC 460 subsystems from overlapping.
- The command is not actually sent until the **CONFIRM** push-button is selected.
- The **CANCEL** push-button can be used to clear the fact that this push-button was selected.



- **HARD RESET**

The **HARD RESET** push-button can be used to reset any controller on the AC 460 subsystem display. Resetting a controller through this push-button has the same effect as depressing the “**Enter**” button on the controller’s front panel.



This button **must not** be used during the on-line upgrade procedure.

The following restrictions apply for this command:

- The command is not actually sent until the **CONFIRM** push-button is selected.
- The **CANCEL** push-button can be used to clear the fact that this push-button was selected.

- **CONFIRM**

Because there is a need to prevent inadvertent selection of a push-button from causing unintended controller resets and switchovers, the **CONFIRM** push-button is used to provide a confirmation process for the “RESET for UPGRADE”, “HARD RESET”, and “SWITCHOVER” functions. The actual reset or switch request is not sent until the **CONFIRM** push-button is selected.

- **CANCEL**

The **CANCEL** push-button is used to clear the fact that any other push-button was selected. You may then **Quit** or select another command button.

### AC410 Subsystem Status Display

The AC410 Subsystem Status Display, [Figure 59](#), provides: node address and name; DCN status; PM150 Controller address, name and status; and TRIO and S100 submodule status and logical LAN numbers. See the AC460 for a description of the TRIO Interface, S100 I/O Interface, S800 I/O Interface, MVI interface and DCN interface.

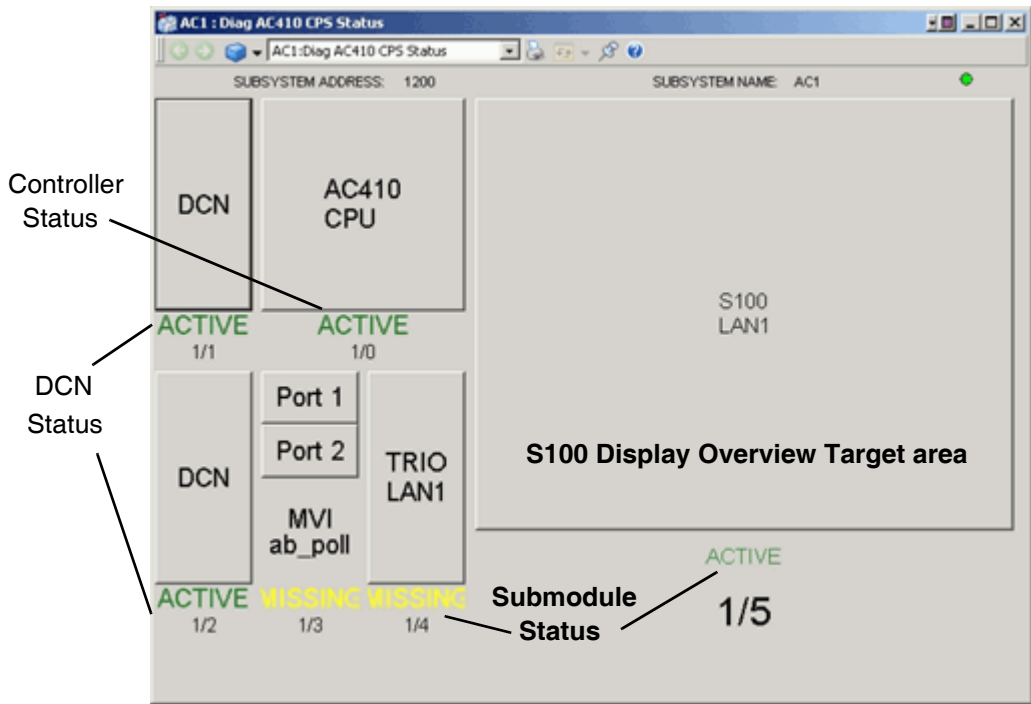


Figure 59. AC410 Subsystem Status Display (Compact Station)

## Controller Node Subsystem Status Display

The Controller Subsystem Status Display, [Figure 60](#), provides information on the specified node's general condition. The first line of the display below the Subsystem Status title provides the device number, type and current state of the Controller Subsystem. The other areas and activities are described below.

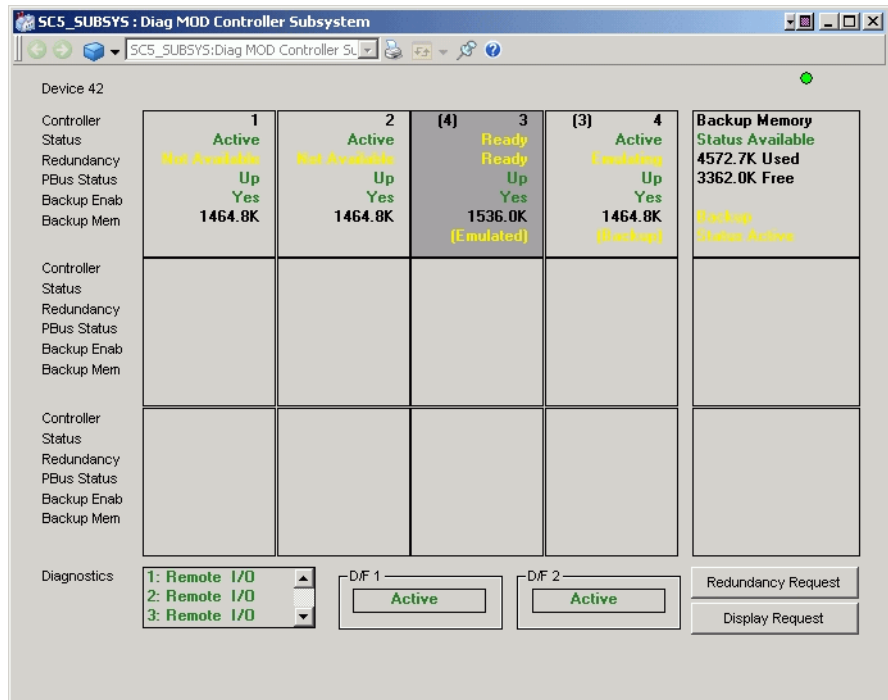


Figure 60. Controller Subsystem Status Display (Controller)

### Controller Subsystem Status Display CCR Requests

The **Redundancy Request** button is used to make Control Card Redundancy (CCR) requests for the selected Control Module. When one of the associated commands is selected, the request and the currently selected subdevice number are shown and a prompt is displayed requesting confirmation of the selected action. If the action and the subdevice are correct, select **Confirm** to send the request. Otherwise, you can select a different command or subdevice. Only when **Confirm** is selected

immediately after the request to confirm is the request executed. You can select **Cancel** at any time to abort a request that has not yet been confirmed.

A Control Card redundancy request is initiated by selecting the status block for the desired Control Module. (The block is highlighted to show selection.) Next, you select the **Redundancy Request** button to bring up a set of commands described below:

**Establish Redundancy** - Establishes redundancy.

**Terminate Redundancy** - Also terminates the Redundancy Auto-Establish mode.

**Start Controller** - Used to get out of the READY state.

**Hard Reset** - Causes the Control Module to reset. This command is similar to pulling the Control Module out of the slot and then re-inserting it.

**Fail Over** - Requests fail over of the Primary Control Module to the Backup Control Module. If the fail over is unsuccessful, the Primary Control Module continues to function as if no request to fail over occurred.

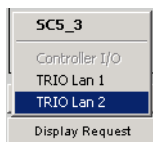
**Switch Back** - Requests switch back from the Backup Control Module to the Primary Control Module.

**Confirm** - Confirms and sends request to the subdevice.

**Cancel** - Cancels any request not yet confirmed.

When one of the above commands is selected (except **Confirm** and **Cancel**), the request and the subdevice number are shown and a prompt is displayed requesting confirmation of the selected action. If the action and selected module are correct, then select **Confirm**. Otherwise, select a different action and/or module.

### Controller Subsystem Status Display Requests



The Display Request button is used to access a **TRIO LAN** display or a Controller I/O (**Direct I/O**) display (only one or the other can be available). First select an available controller, then select the Display Request button and then select the appropriate display. Alternately, you can select a specific Remote I/O or Direct I/O in the Diagnostic area.

## Controller Subsystem Status Display Description

The main areas on the Controller Subsystem Display are:

### CONTROLLERS

Status and redundancy information for up to 12 Control Modules are shown in individual status blocks whose arrangement corresponds to the slots in the Controller Card File(s) of the subsystem. If the subsystem employs 3-1 controller redundancy, blocks 4, 8 and 12 show the status of the backup controllers and all other blocks show the status of the primary controllers. If the subsystem employs 11-1 controller redundancy, block 4 shows the status of the backup controller and all other blocks show the status of the primary controllers.

A blank status block indicates that there is no Control Module configured for that slot in the Controller Card File. If a status block contains only the status, data in the Control Module itself could not be accessed. Each field in a status block is a target. When selected, it is highlighted and that Control Module becomes the object of subsequent redundancy and display requests. If a target in the status block for a Backup Control Module which is emulating a Primary Control Module, or a Primary Control Module which is being emulated by a Backup Control Module is selected, the subdevice ID is the modules current pseudonym. (For example, if Control Module 4 is emulating Control Module 3, their subdevice IDs are 03 and 04, respectively.)

The controller status is shown in the fields of its respective status block as:

### STATUS

- ?????? - No status received (Red)
- UNKNOWN (Red)
- DOWN (Red)
- READY (Yellow)
- IMPAIRED (Yellow)
- ACTIVE (Green)
- AVAILABLE (Cyan)

### REDUNDANCY

- ?????????? - No status received (Red)
- UNKNOWN (Red)
- FAILED (Red)

- EMULATING - Back-up controller only (Yellow)
- NOT AVAIL (Yellow)
- LOST (Yellow)
- TERMINATED - Controller only (Yellow)
- READY (Yellow)
- ACTIVE (Green)
- EMULATED - (Yellow)

After redundancy has been LOST, Control Modules attempt to re-establish redundancy every 30 seconds. This operation is similar to the establish redundancy request except no diagnostic is generated. The redundancy state remains LOST for five unsuccessful attempts. After the fifth unsuccessful attempt to re-establish redundancy, the redundancy status becomes TERMINATED. If redundancy is re-established before the fifth attempt, the REDUNDANCY AUTO-ESTABLISH AFTER x TRIES diagnostic message is generated. Once the auto-establish mode is entered, it remains active even if a manual **Redundancy Request > Establish Redundancy** request causes a transition to the FAILED state. It can be stopped by selecting **Redundancy Request > Terminate Redundancy**.

#### PBUS STATUS

- ??????? - No status received (Red)
- DOWN (Red)
- UNKNOWN (Red)
- UP (Green)

#### BACKUP ENAB

- NO (Yellow)
- YES (Green)

#### BACK-UP MEMORY

Status information for up to three Backup Memory Modules per Controller Subsystem are shown in individual status blocks to the right of the status blocks for the controllers they back up. Each field in a status block is a target. When selected, it is highlighted and that Backup Memory Module becomes the object of subsequent redundancy and display requests. (The highlighted backup controller status block(s) in the CONTROLLERS area indicate(s) which set or sets of up to four controllers are backed up by that backup memory module.) Backup Memory Modules are assigned pseudo-subdevice numbers of 13 (OD),

14 (OE) and 15 (OF) since they are not actually on the DCN or F-Bus. The only meaningful displays for use with Backup Memory Modules are the Diagnostic Message Displays.

The backup memory status is shown in the status block as:

#### STATUS

- ?????? - Unknown (Red)
- UNKNOWN (Red)
- NOT AVAIL (Red)
- ACTIVE (Green)

#### BACKUP STATUS

- ?????? - No status received (Red)
- FAILED (Red)
- UNKNOWN (Red)
- ACTIVE (Yellow)
- READY (Green)

#### D/F

Each of these two fields is a target. When selected, it is highlighted and that D/M Module becomes the object of subsequent redundancy and display requests. D/M Modules are assigned pseudo-subdevice numbers of 30 (1E) and 31 (1F), rather than the subdevice number of 0 which is indicated on the System Status Display, in order to distinguish messages from each D/F. The only meaningful displays for use with D/F Modules are the Diagnostic Message Displays.

The D/F status, shown in the D/F fields, is either DOWN (Red) or ACTIVE (Green).

#### DIAGNOSTICS

The status of power up diagnostics for each module accessible on the DCN is reported in this field. This area is always the Page Forward and Page Back region.

The possible diagnostic errors for the Control Module, Backup Memory Module and D/F Module are:

- RAM Error (fatal error)
- PROM Checksum Error (fatal error)

- Timer Error (timer is not within a set tolerance during a 10 ms delay)
- Watchdog Timer Error (on-board watchdog timer times out in under 350 ms or takes longer than 410 ms)
- Bus Error Handling Error (system getting bus errors on valid addresses or not getting errors on invalid addresses)
- Parity Handling Error (system is not correcting a one bit error or not detecting a two bit error)
- D/M Error (M/F cannot communicate with the D/M or the D/M has an error)
- XMEM Not Supported
- Unknown Error
- Analog Output Error
- Analog Input Error (error reading high or low reference channel or standard channel)
- Digital Input Error (error reading high input, low input, or unable to get interrupt)
- Digital Output Error

With the exception of diagnostic detection of an output channel failure, all of the above failures to a Control Module automatically initiate the fail over process. A failed output channel is configured to perform one of the following four actions based upon a detected diagnostic failure:

- Continue processing the loop as configured
- Turn the Output FCM off
- Fail over
- Turn the loop off

If, after a fatal diagnostic error, fail over is unsuccessful, either because redundancy is not available or the switch to back-up did not occur, power to the outputs of the failed Control Module is turned off. This is accomplished by a hardware timer that resets the outputs approximately 200 milliseconds after the card available signal is turned off. If in the HOLD position, the last outputs are maintained. Otherwise the output values go to zero. If a failure is the result of a shorted transistor on the digital output, transfer to the Backup Control Module, whose I/O is paralleled with the Primary, results in the Backup failing also. This causes the Backup to reset and output power is turned off.

Diagnostics detect if a relay in the analog signal path fails, but is unable to detect the failure of a relay in the digital signal path.



## Turbo/Console Node Subsystem Status Display

The Turbo/Console Node Subsystem Status Display, [Figure 61](#), provides information on the specified node's general condition. The first line of the display below the Subsystem Status title provides the device number, type and current state of the Multibus Subsystem. The other areas and activities are described below.

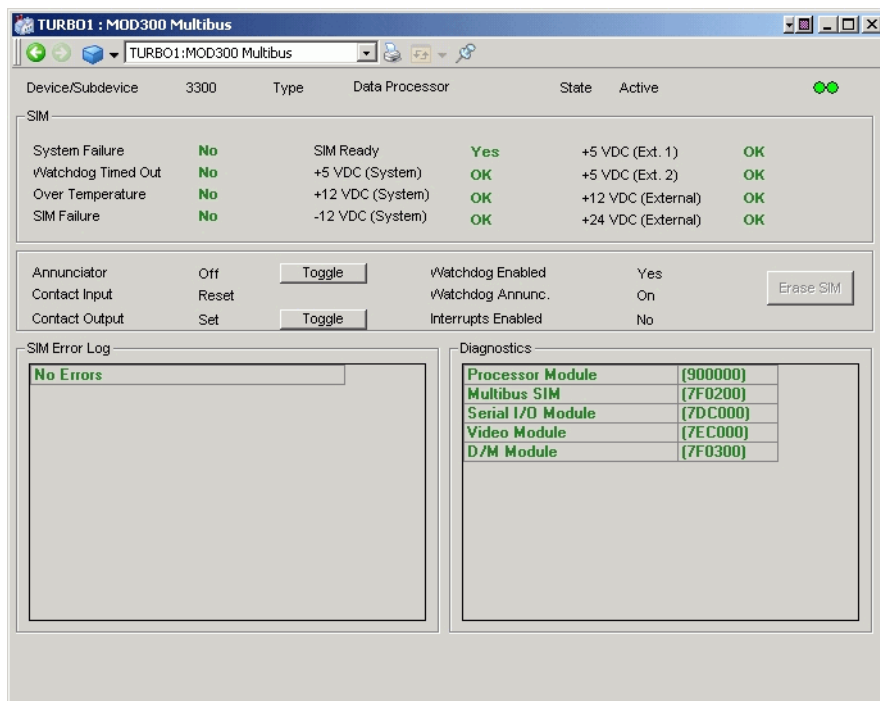


Figure 61. Turbo Node Subsystem Status Display (Data Processor)

### SIM

The current status data, reported by the System Integrity Module (SIM) for the specified subsystem, is shown in this field. Buttons are also available for controlling the Annunciator, the Contact output to erase the SIM.

## SIM ERROR LOG

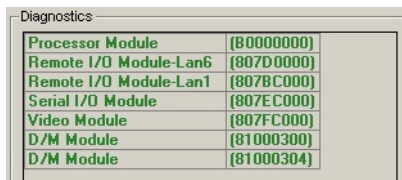
The current error log, as reported by the System Integrity Module for the specified subsystem, is shown in this field. Note that only fatal errors are shown. The hexadecimal number to the left of the colon is a unique hardware, software and SIM status code identifier. The set of four hexadecimal numbers to the right of the colon describes the specific type of error associated with the unique status code identifier. Refer to your *Subsystem Status and Diagnostics* book for descriptions of the codes and their meanings.

## DIAGNOSTICS

The status of power up diagnostics for each module in the subsystem is reported in this field. A description and the I/O address of each module found in the subsystem is provided. Refer to your *Subsystem Status and Diagnostics* book for a list of the possible I/O module base addresses.

When the System Integrity Module (SIM) is missing or not active, the message "NOT AVAILABLE" is displayed in the SIM and SIM ERROR LOG areas.

When the subsystem includes remote I/O, [Figure 62](#), then Remote I/O rows show the LAN number. This number is the one configured in the node not in the database.



Diagnostics	
Processor Module	(80000000)
Remote I/O Module-Lan6	(807D0000)
Remote I/O Module-Lan1	(807BC000)
Serial I/O Module	(807EC000)
Video Module	(807FC000)
D/M Module	(81000300)
D/M Module	(81000304)

Figure 62. Turbo Node Subsystem Status Diagnostic Area with TRIO

## Subsystem Status Display Operation

Buttons available and the actions performed by each include:

<b>ANNUNCIATOR</b>	Toggles the SIM Annunciator state ON or OFF.
<b>CONTACT OUTPUT</b>	Toggles the SIM Contact Output state SET or RESET.
<b>ERASE SIM</b>	Clears current error memory locations 01 through C8 on the SIM.

## Diagnostic Message Display

The Diagnostic Messages Display, [Figure 63](#), shows all messages reported and filed as a diagnostic failure by a device/subdevice of the MOD control system (both system software and system hardware) along with an indication of the time, frequency and type. A font selection allows you to display large, medium or small fonts and consequently fewer or more lines in the main portion of the display. Colors are assigned to each message type as defined below:

Color Indication	Diagnostic Message Type
Red	Fatal (F) and Error (E)
Yellow	Warning (W)
Green	Informational Status (S)

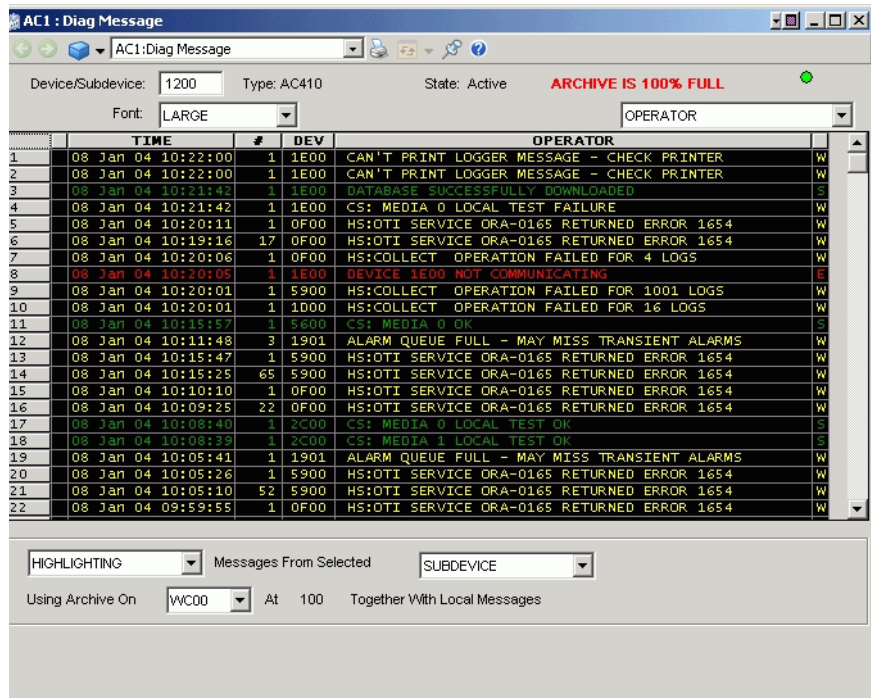


Figure 63. Diagnostic Message Display

## Device Identification

The first line of the Diagnostic Message Display provides the device/subdevice number, type, and state of the device.

## Archive Messages

The total number of records stored in the disk archive file is shown as a percentage of the total size of the archive in the MESSAGE mode, or as an exact number of messages in the ARCHIVE mode. Regardless of the mode, the number of records is shown in yellow if the file is 75 to 90 percent full and red if over 90 percent full. This serves to indicate when the oldest messages in the archive are in danger of being lost. This occurs because the archive has a fixed size, and if it is full (2999 messages) the oldest message is erased to make room for a new message. It is not necessary to do anything about this condition unless there have been problems which require future examination of messages that are now in the archive.

## Diagnostic Messages/Archive Display Description

The main fields on the Diagnostic Messages/Archive Display are:

- TIME** Shows the date and time of the most recent occurrence.
- #** Shows the number of occurrences within an interval of up to 10 minutes.
- DEVSUB** Shows the device/subdevice ID from which the message originated.
- DESCRIPTION** Shows the message text and severity. The first time a message occurs, it is archived with a frequency count of 1 and saved in local memory. Each time the same message occurs again, the frequency count increments in the local copy. On the display, indication that a message is being counted in local memory is shown by a greater than sign (>) in the first column. Ten minutes after the first occurrence, or when flushed from local memory to make room for newer messages, the message is archived with its final time and count (if more than one) and removed from local memory. If the same message occurs again, the process starts over. The severity of each message is indicated by the letter to its right and by a color as follows (default colors indicated):
- F (Fatal) Red - A fatal error occurred to the node (which causes fail over in Control Modules).

- E (Error) Red - An error occurred which prevents an individual package from running.
- W (Warning) Yellow - An abnormal event occurred which does not prevent anything from running.
- S (Success) Green - A procedure completed successfully.

## Diagnostic Messages Display Targets

Additional fields/targets available on this display and their use are:

**OPERATOR** By default, OPERATOR messages are displayed (those which cause the DIAG target to flash). To change the message format, use the pull-down menu to select either OPERATOR, ALL MSGS - TEXT, ALL MSGS - CODE, and ALL MSGS - ALT. In the ALL MSGS formats, all messages are displayed, including minor problems, which may have led up to an actual failure. In the ALL MSGS-TEXT format, AP error codes are translated into text (as is done for operator messages). In the ALL MSGS-ALT format the "fixed" error code is translated into text indicating the operation being performed by the package that generated the message. In the ALL MSGS-CODE format, all codes are given numerically.

If you need to contact Technical Service for help, it is important to provide them with as much information as possible concerning the observed problem and actions attempted. The ALL MSGS - CODE, for example, identifies the AP code that can be found in the *Diagnostic Error Messages* book to determine the cause and effect of a problem as well as the action to be taken.

**DEVICE/SUBDEV** The current device/subdevice ID (in the upper left) indicates for which node messages are displayed or highlighted. The subdevice ID also indicates the subdevice for which data is shown on subsequently called up displays. To view another Device/Subdevice, select the appropriate object and use the context menu.

**EMPHASIS MODE** The emphasis mode determines whether messages from the selected source are the only messages shown (SHOWING ONLY) or are highlighted to distinguish them from others (HIGHLIGHTING). To change the emphasis mode, use the pull-down menu to select either the SHOWING ONLY and HIGHLIGHTING modes. The messages displayed or highlighted are further limited by the emphasis source. The effect of the emphasis mode and source is described below.

**EMPHASIS SOURCE** The emphasis source determines whether the set of messages for the current device/subdevice (SUBDEVICE), subsystem (SUBSYSTEM), or archive file (ARCHIVE) are displayed or highlighted. To change the emphasis source, use the pull-down menu to select either the SUBDEVICE, SUBSYSTEM, and ARCHIVE source. The messages displayed or highlighted are further specified by the archive disk described below.

**DISK NAME** The disk identified in this field is the previously recorded archive disk from which messages are being viewed. To change the archive disk, use the pull-down menu to select either the available disk from which you want to view messages. If an archive other than WC00 (the configurator disk) is selected, current messages from the selected subdevice are not shown to avoid confusion between recent messages on this system and previously archived messages from this or another system.

The EMPHASIS MODE and EMPHASIS SOURCE fields together select six possible ways to view messages. The effects are summarized below:

MODE	SOURCE	EFFECT
SHOWING ONLY	SUB-DEVICE	Displays current (if the disk is WC00) and archived messages from the selected device/subdevice. The messages shown are the same as those shown in the default MESSAGES mode.
SHOWING ONLY	SUB-SYSTEM	Displays archived messages from all subdevices of the selected device, highlighting messages from the selected subdevice.
SHOWING ONLY	ARCHIVE	Displays archived messages from any device. The messages shown are the same as those shown in the ARCHIVE mode.
HIGH-LIGHTING	SUB-DEVICE	Displays archived messages from any device, highlighting those from the selected device/subdevice.
HIGH-LIGHTING	SUB-SYSTEM	Displays archived messages from any device, highlighting those from any subdevice of the selected device.
HIGH-LIGHTING	ARCHIVE	Displays current and archived messages from the selected device/subdevice, highlighting archived messages.

## Printing Diagnostic Messages

The print view icon along the tool bar prints a screen dump of the diagnostic message display. To print selected messages, use the **Print Diagnostic Messages** context menu selection on the Diagnostic Message Display. Use the standard print setup function to select a printer. After selecting the print view icon, the print display, [Figure 64](#), is shown.

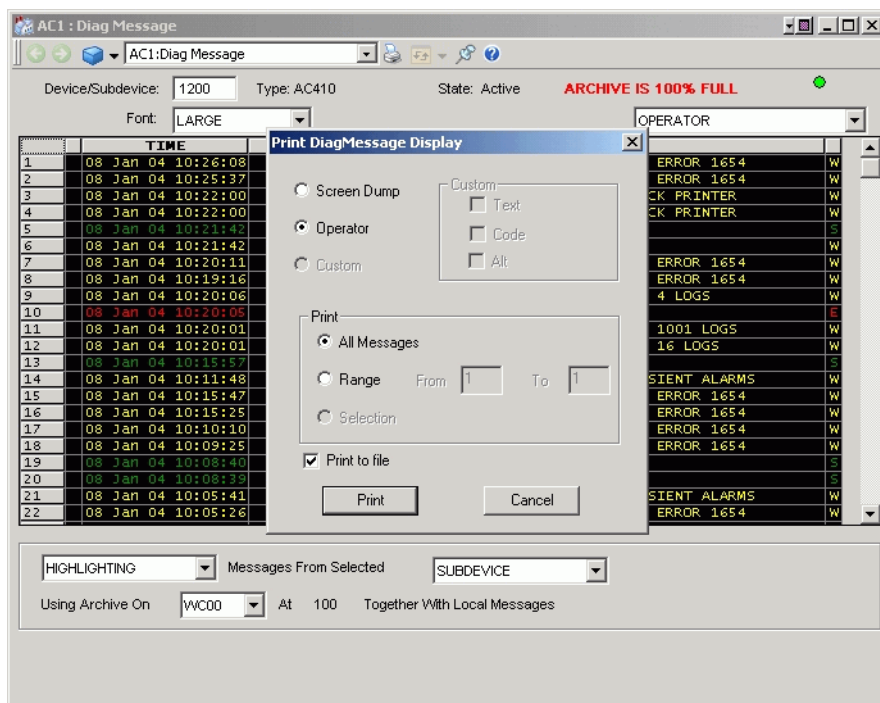


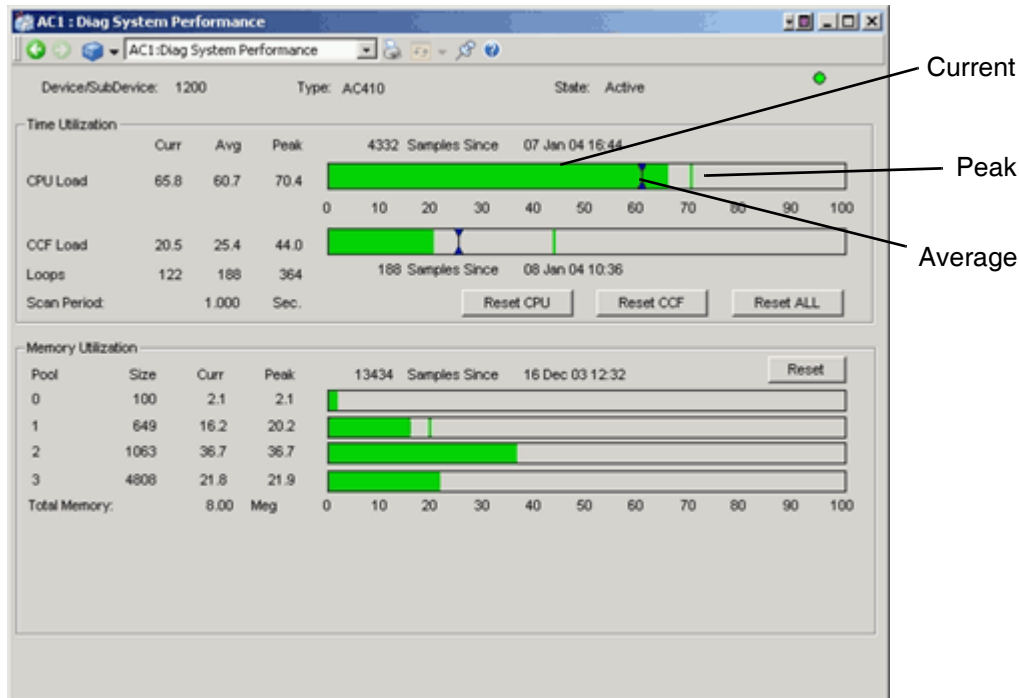
Figure 64. Printing Diagnostic Messages

Select **Screen Dump** to print a picture of the current view. Use the Operator or Custom selection to print All Messages or a **Range** of messages where the range numbers are row numbers. Custom mode is selectable when the message format is not OPERATOR.

The ability to Print to file is also included as a way to send messages to a text file or other print format.

## System Performance Display

The System Performance Display, [Figure 65](#), is called up from the Subsystem Status Display. It provides information on the performance of a specified device/subdevice (node) in terms of CPU loading, Configurable Control Functions (CCF) software loading, and memory pool usage.



*Figure 65. System Performance Display*

The display provides information on the performance of a specified device/subdevice (node) in terms of CPU loading, CCF software loading and memory pool usage. There are three performance monitors on this display, each containing one or more bar graphs showing percent usage. In each graph, current value is indicated by a solid bar, peak by a yellow line, and average (where used) by a blue marker.



The bar graphs are color coded. Red indicates a high level of resource utilization (90 to 100%), yellow a moderately high level of utilization (75 to 90%), and green a moderate level of utilization (0 to 75%). White is used to indicate neutral facts.

Excessive CCF load (red region) may be caused by problem areas such as:

- The BASERATE field on the CCF Templet causing high average usage.
- The PROCESSING RATE or PROCESSING PHASE field on the Loop Definition or Device Loops Templet causing large peaks and low current usage.
- Continuous or device loops driven by digital inputs causing high current and average usage.
- Alarm frequency due to narrow deadbands (causes high peak usage).

## Reset ALL

Resets the "samples since" time year and month. YYYYMM.0 for **Reset CPU**, **Reset CCF**, and **Reset** (Memory utilization).

## Reset CPU

The **Reset CPU** button resets DATA1 of a SYS Data Entry FCM to zero which resets the CPU load "samples since" time year and month.

## Reset CCF

The **Reset CCF** button resets DATA7 of a SYS Data Entry FCM to zero which resets the CCF load "samples since" time year and month.

## Reset Memory Utilization

The **Reset** button resets DATA1 of a MEM Data Entry FCM to zero which resets the memory utilization "samples since" time year and month.



---

## Section 7 I/O Displays

### I/O Displays Overview

I/O displays contain information that allow you to identify detectable fault conditions, monitor general status and performance, change process outputs directly for testing, and control the status of redundancy. The I/O displays include:

S800

S100

PROFIBUS

TRIO

Direct I/O

Entries identifying an operator action on the I/O displays are relayed to the appropriate logging device. Color coding of data shown on the displays is like the diagnostic displays. Hardware factors to consider are:

- No more than 30 TRIO blocks can be attached to a Field Bus Controller.
- A SC Controller I/O Module has either all TRIO or all Direct I/O.
- A SC Controller can have two Field Buses at most. Each can be either non-redundant or redundant.
- All SC Controllers backed up by one specific backup controller must have the same type of I/O: either TRIO or Direct I/O.
- An Advant Controller 460 can have a mixture of TRIO and local S100 I/O, with a maximum of four non-redundant or two redundant Field Buses.
- An Advant Controller 410 can have a mixture of TRIO and local S100 I/O, with a maximum of two non-redundant or one redundant Field Bus(es).

## S800 I/O Displays

These S800 runtime displays relate to the Series 800 process interface as described in the *S800 I/O User's Guides* where you can find additional information about S800 I/O.

### S800 LAN Display

The S800 LAN display, [Figure 66](#), shows the CI520/CI522 submodule status and information about each S800 I/O Station configured under a particular LAN. Select a display target to call up the [S800 Station](#) display.

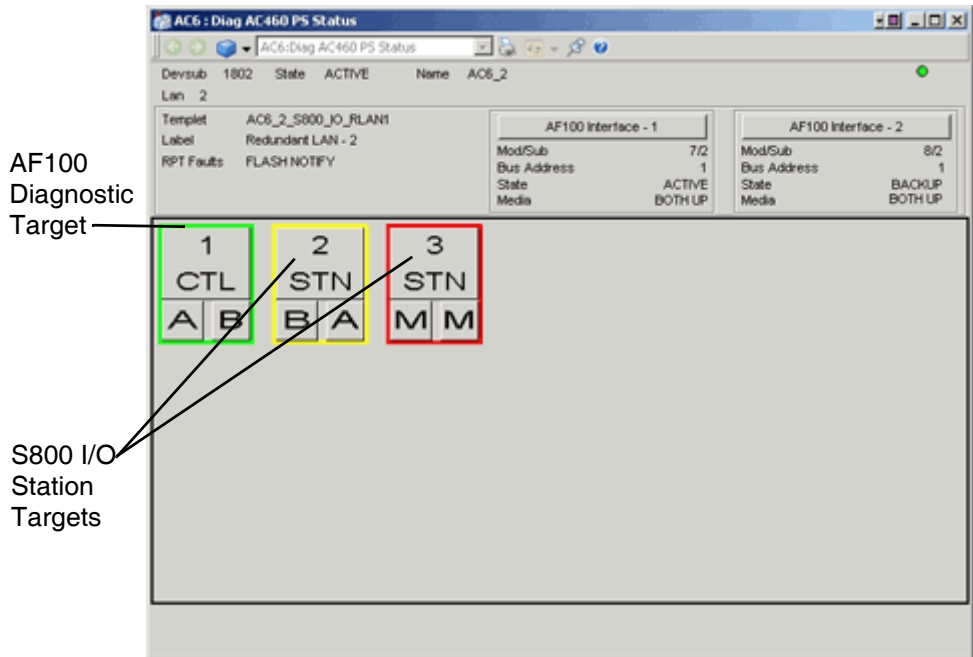


Figure 66. S800 LAN (Overview) Display

The heading fields on the S800 LAN display are described below.

<b>Devsub</b>	The controller DCN address where the S800 I/O is located.
<b>State</b>	This field shows the status of the controller as ACTIVE, DOWN or AVAIL.
<b>Name</b>	The controller subsystem part of the logical name created during database configuration.
<b>Lan</b>	This field represents the number of the Local Area Network (LAN) for all the stations shown. For the AC 460 this is LAN # 1 - 4 and for the AC 410 this is LAN # 1 or 2. Go back to the controller display to select a different LAN.
<b>Templet</b>	The logical device name created during database configuration is given in this field.
<b>Label</b>	Descriptive label associated with templet.
<b>RPT Faults</b>	Indicates the reporting method for fault condition. FLASH NOTIFY, SILENT NOTIFY or NO NOTIFY.

#### **AF100 INTERFACE - X Status Area**

The AF100 Interface status area(s) represent the CI520 AF 100 Interface or the redundant CI522 AF 100 Interfaces and are described below. For the CI520, the box contains the current state of the AF100 INTERFACE - 1. For the CI522, the left box contains the current state of the AF100 INTERFACE - 1 (as defined in the LAN templet), and the right box contains the current state of the AF100 INTERFACE - 2.

<b>Mod/Sub</b>	This field represents the slot number of the of the controller carrier MODule, and the SUBmodule location of the CI520/CI522.
<b>Bus Address</b>	This is the AF 100 field bus address assigned to the CI520/CI522.

- State** This field shows the status of the submodule.  
 ACTIVE indicates the submodule is in use.  
 BACKUP indicates the backup submodule is OK and ready.  
 FAILED means the submodule has stopped.  
 MISSING means there is not a submodule installed in this location.
- Media** This shows the redundancy status of the AF 100 bus media.  
 NONE means that no bus is connected.  
 BOTH UP means that dual media is used and operational.  
 CABLE 1 UP or CABLE 2 UP shows if only one bus is connected and operational.

### AF100 INTERFACE - X Menu

Each AF100 Interface status area contains an AF 100 INTERFACE - X menu button. When the AF100 INTERFACE - x button is selected, a menu is displayed with the items shown in [Figure 67](#) and as explained below.

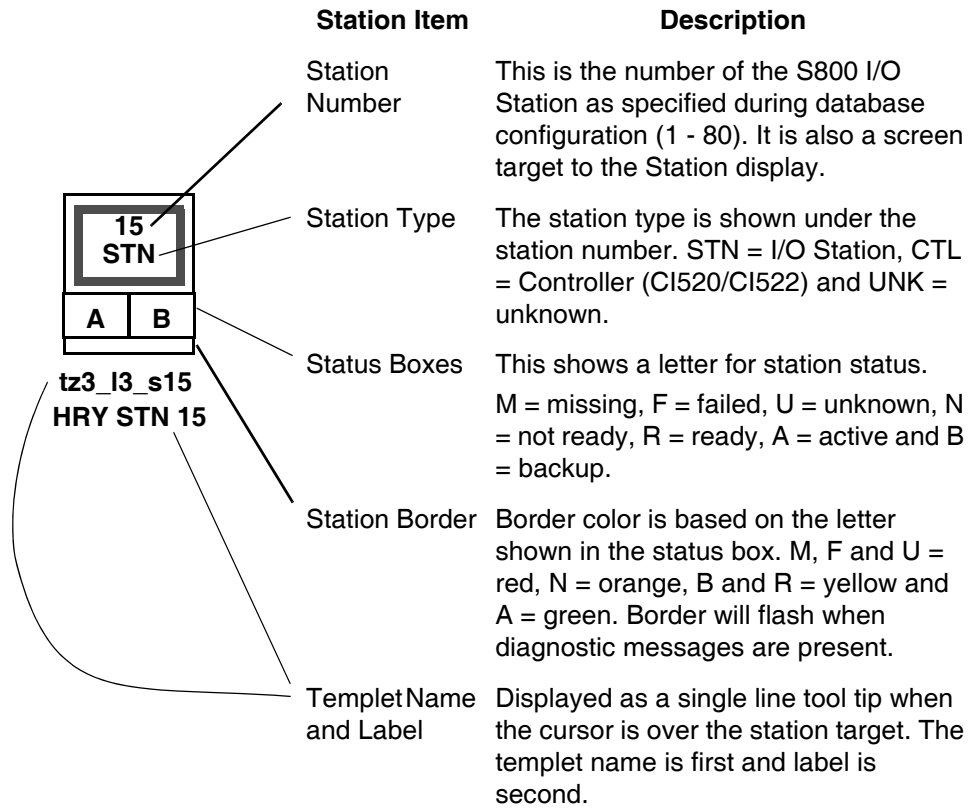


Figure 67. AF 100 INTERFACE - X menus

- State** User can change state to:  
 RESTART re-initializes the lan,  
 DEACTIVATE forces all data qualities to 'bad',  
 ACTIVATE sets data qualities based on the active status.  
 SWITCHOVER switches to a backup interface if present.
- RPT Faults** User can select to change reporting method: Flash Notify, Silent Notify or No Notify.
- Get Diagnostics** Select this to get the diagnostics dialog, [Figure 68](#).

**S800 Station Detail**

In the main section of the display, a detail window shows specific information about each I/O Station. The right scroll bar is used to view other stations on the LAN. The display is arranged in a 10 by 8 grid for all stations (1 - 80). Stations not configured are not shown. Station 80 being in the lower right corner can be seen by scrolling the window to that area of the grid. For each I/O Station, information is displayed as described below.



If the station is the CI520/CI522 type CTL, selecting that station icon will bring up the CI520/CI522 Diagnostics Display, [Figure 68](#).

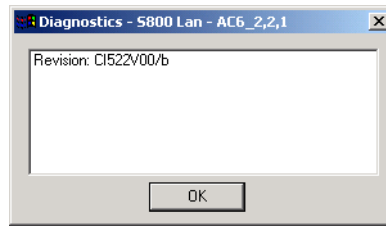


Figure 68. CI520/CI522 Diagnostics Display

### S800 Station Display

The S800 Station display is started by selecting a configured I/O station from the [S800 LAN Display](#). The S800 Station display, [Figure 69](#), shows information about the FCI Module, each S800 I/O Module and the I/O Clusters that are configured for the I/O Station.

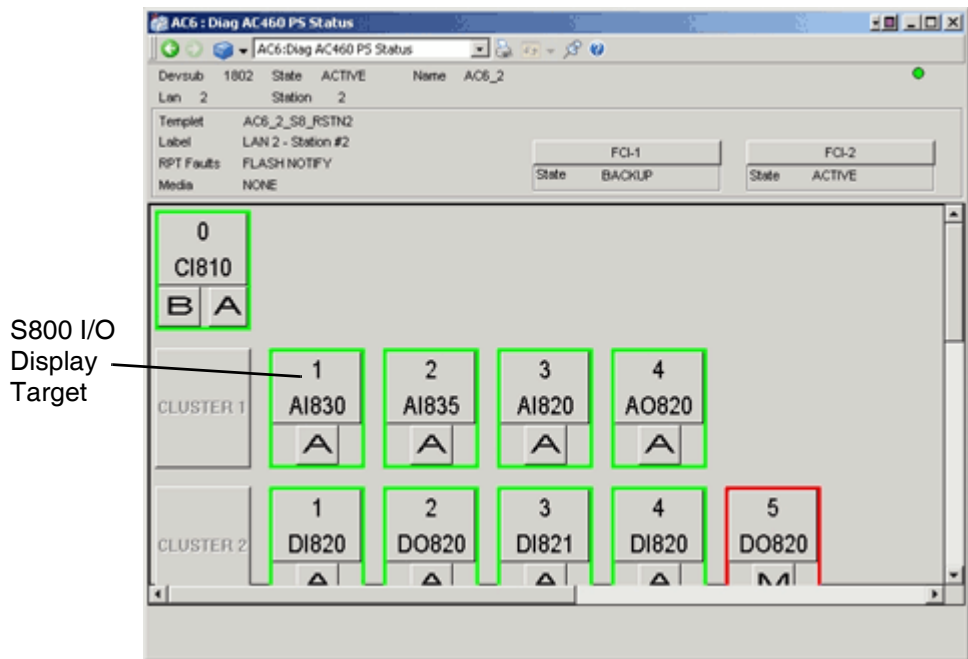


Figure 69. S800 Station Display



Select an I/O Device display target to get a [S800 Device Display](#). Use the context menu to move up to the LAN display. Select the CI810 (FCI) box to get the CI810 Diagnostics Display (similar to [Figure 68](#)).

The top of the S800 Station display identifies the I/O Station, its Advant Controller and has other information as described below.

<b>Devsub</b>	The controller DCN address where the S800 I/O is located.
<b>State</b>	This field shows the status of the controller as ACTIVE, DOWN or AVAIL.
<b>Name</b>	The controller subsystem part of the logical name created during database configuration.
<b>Lan</b>	This field represents the number of the Local Area Network (LAN) for all the stations shown. For the AC 460 this is LAN # 1 - 4 and for the AC 410 this is LAN # 1 or 2. Go back to the controller display to select a different LAN.
<b>Station</b>	This field represents the number of the station. Go back to the LAN display to select a different station.
<b>Templet</b>	The logical device name created during database configuration is given in this field.
<b>Label</b>	Descriptive label associated with templet.
<b>RPT Faults</b>	Indicates the reporting method for fault condition. FLASH NOTIFY, SILENT NOTIFY or NO NOTIFY.
<b>Media</b>	This shows the status of the AF 100 bus media. NONE means that no bus is connected. BOTH means that dual media is used and that both are operational. CABLE 1 UP or CABLE 2 UP shows if only one bus is connected and operational.

### FCI - X Status Area

FCI Status areas represent a CI810A FCI or the redundant CI820 FCIs. For the CI810A, the status box contains the current state of FCI - 1. For the redundant CI820 FCIs, the left status box contains the current state of FCI - 1 (or the left FCI), the right status box contains the current state of FCI - 2 (or the right FCI).

<b>State</b>	This field shows the status of the FCI (CI810A/CI820). ACTIVE indicates the FCI is in use. BACKUP indicates the station is running redundantly and the backup is available. READY indicates a transient state on the way to becoming ACTIVE or BACKUP. FAILED means the FCI has stopped. MISSING means there is not an FCI connected. UNKNOWN is an indeterminate state, sometimes seen as a transient state as the station changes state.
--------------	--

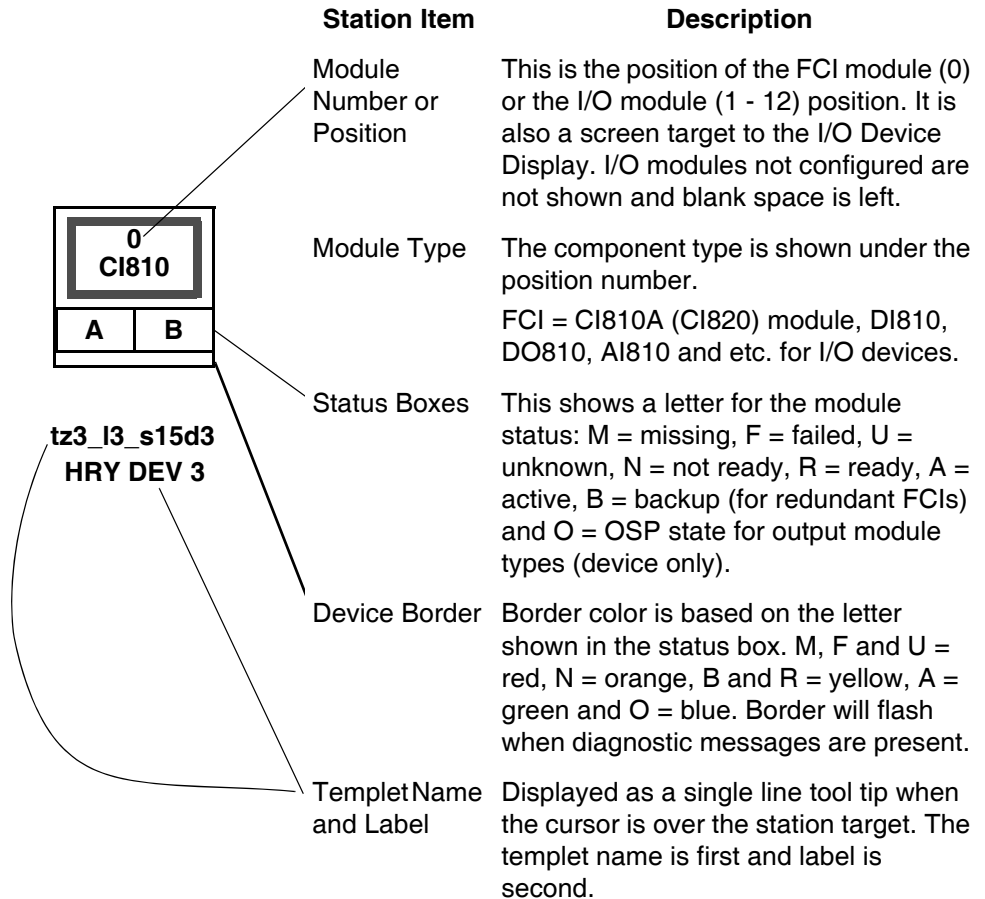
### FCI - X Menu

Each FCI - x status area contains a FCI button. Select the FCI button to display the menu items described below.

<b>State</b>	User can change state to: RESTART re-initializes the lan, DEACTIVATE forces all data qualities to 'bad', ACTIVATE sets data qualities based on the active status. SWITCHOVER switches to a backup interface if present. Switchover is not used for non-redundant FCIs.
<b>RPT Faults</b>	User can select to change reporting method to: Flash Notify, Silent Notify or No Notify.
<b>Get Diagnostics</b>	Select this to get the diagnostics dialog.

**S800 Device and FCI Detail**

In the main part of the display, [Figure 69](#), a detail window shows specific information about the FCI, I/O Clusters and each I/O module device. The right scroll bar is used to view other devices of the station. For the FCI and each I/O module, information is displayed as described below (see [Figure 69](#) for device graphic, FCI shown below).



## S800 Device Display

The S800 Device display is started by selecting a configured I/O device from the S800 Station Display. The S800 Device display, [Figure 70](#), shows information for the device and each channel configured for the S800 I/O module (device diagnostics are not displayed). Use the context menu to move up to the station and LAN displays. From this display you can:

- Change channel output value
- Select loop CCF tag and start loop faceplate
- Select TLL point tag and start TLL faceplate
- Select device configuration display

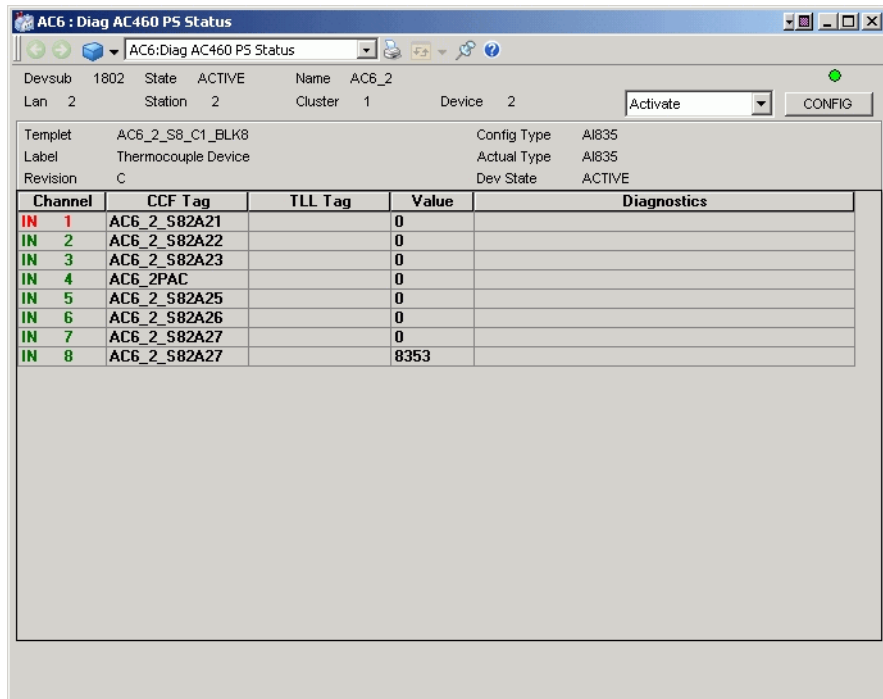



Figure 70. S800 Device Status Display

At the top of the display there is a section related to the addressing of the I/O module.

Module Address Info	Description
<b>Devsub</b>	This field is the controller DCN address where the S800 I/O is located.
<b>Name</b>	This field shows the name of the AC460 Controller Subsystem.
<b>State</b>	This field shows the status of the controller as ACTIVE, DOWN and AVAIL.
<b>Lan</b>	This field represents the number of the Local Area Network (LAN) of this station.
<b>Station</b>	This field represents the number of the I/O Station.
<b>Cluster</b>	This field represents the cluster number of the I/O Station. Cluster 0 (zero) is the I/O modules connected directly to the FCI. Clusters 1 - 7 are for remote optical ModuleBus connections to the FCI.
<b>Device</b>	This field represents the device number of the I/O module of the cluster selected on this station.
Command Menu	Set a device condition by pointing to the COMMAND drop down menu and selecting the desired action:
	<p>RESTART re-initializes the device,</p> <p>DEACTIVATE forces all data qualities to 'bad' (used if applications need to be informed while the device is not ready to be used, for example, if maintenance is being performed),</p> <p>ACTIVATE sets data qualities based on the active status of each channel.</p>
<b>Config</b>	Brings up the I/O Module Configuration Display (AI810/AI890, AO810/AO890, DI810, DI820, DO810, DO820), which gives the configuration for that module and allows configuration changes. See <a href="#">S800 I/O Runtime Templates</a> on page 285.

The next section gives information about the I/O module selected.

<b>Templet</b>	The device name created during database configuration is given in this field.
<b>Label</b>	This field shows the label given this I/O module.
<b>Revision</b>	Gives the revision level of the module if available.
<b>Config Type</b>	This field shows the I/O module type that was configured for this device number.
<b>Actual Type</b>	This field shows the actual I/O module type that is inserted in this device number location.
<b>Dev State</b>	This field shows the status of the I/O module. ACTIVE indicates the I/O module is OK. FAILED means the I/O module has stopped. MISSING means there is not an I/O module installed.

The channel window section shows specific information about each channel of the module. A scroll bar is used to view other channels of the device. For each channel, several columns of information are displayed as described below.

<b>Channel</b>	<p>The input (IN) or output (OUT) and channel number of the device. The text color for each channel is based upon the reporting of faults and the data quality:</p> <ul style="list-style-type: none"> <li>Dark Green = Report faults is yes and data quality is good.</li> <li>Red = Report faults is yes and data quality is bad</li> <li>Light Green = Report faults is no, data quality is forced to 'good'</li> </ul>
<b>CCF Tag</b> and <b>TLL Tag</b>	<p>The TAG is the name of the CCF loop or the TLL device that uses the channel. If no loop or device is assigned to the channel then the field is blank.</p> <p>Select the CCF or TLL Tag target area to get the appropriate MOD Faceplate. If a tag is configured for warmstart and is warm started, it is shown with a white background.</p>

<b>Value</b>	<p>The actual value of the input from the process or the output value sent to the process.</p> <p>For an output field, you can change the value of an output channel by selecting it and entering a value. the loop needs to be warm-started from the FCM display.</p>
<b>Diagnostics</b>	<p>Not used.</p> <p>See the S800 I/O instruction for error message information.</p>

## S800 Warmstart

### Actions Upon Startup

Upon startup, outputs to all devices are disabled. The database is downloaded with each CCF loop output FCM in its initial output mode and each device loop holding its initial command. A period of time is allowed for each S800 module to log into the system. As each module is recognized, its outputs are read. If configured for warmstart, the corresponding CCF loop output FCM for each output channel on the module is placed in the MANUAL mode and its result is synchronized with the field output value in the module memory. The action taken by a device loop configured for warmstart depends on whether there are inputs to the device. If there are inputs to the device, the DEV\_CMND parameter is adjusted to command the device to assume the existing field state. If there are no inputs, the DEV\_CMND parameter is set to the INIT\_CMND (NO COMMAND). Finally, the automatic transfer of outputs to the S800 module is enabled.

### Actions On Module Communication

As a module logs out, a diagnostic message “Block nnnn has logged out” is sent and the associated data quality goes BAD. As a result, FCMs may take their configured “action on output error” and the automatic transfer of outputs to the module is disabled. As a module logs in, output values are read from the module. If configured for warmstart, the corresponding CCF output FCM or device loop for each channel on the module is warm started. That is, the output FCM is placed in the MANUAL mode, and the output value from the module is written to its result attribute. For the warm started device loop, its DEV\_CMND attribute is set to NO COMMAND. Finally, the automatic transfer of outputs to the S800 module is enabled.

### **Actions On Loop Download**

When a loop with an output FCM configured for warmstart is downloaded, the output mode of the FCM is placed in the MANUAL mode and the output value from the module is written to its result attribute. When a device loop configured for warmstart is downloaded, its DEV\_CMND attribute is set to NO COMMAND.

### **Runtime Display Support**

During runtime, but not at startup, as modules log in, diagnostic messages are sent to the diagnostic archive. On the S800 I/O Display, there is a cross reference for a CCF tag associated with each channel. On this display, the tags for CCF loops with output FCMs and device loops configured for warmstart that were warm started are shown with a white background. Note that the loop warm started for a given channel is the loop whose tag is shown in the S800 I/O Display. If two or more loops reference the same channel, only the last loop downloaded participates in warmstart and is listed on the S800 I/O Display.

### **Operation Actions**

With warmstart configured for output FCMs and device loops, the authorized user should:

- acknowledge device acquisition diagnostics (at node startup, it is assumed all modules present are warm started)
- call up the corresponding S800 I/O Display to determine which tags are affected by warmstart
- make any necessary modifications to synchronize the control system with the current outputs
- commission affected CCF output FCMs by setting their modes to AUTO
- commission affected CCF device loops by issuing device commands (if desired)





For continuous loops with Analog and Digital Output FCMs or device loops configured for warmstart that have been warm started, the tag is shown with a white background.

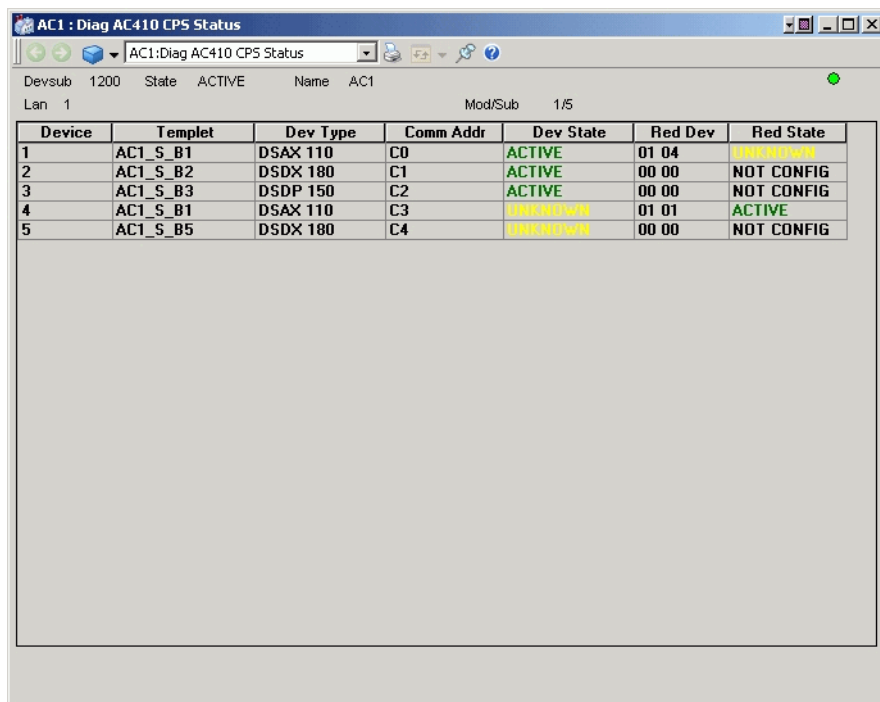
When CCF loops with output FCMs configured for warmstart are placed in the MANUAL mode by either a warmstart or user action, their tags are shown with a white background. When device loops configured for warmstart have their DEV\_CMND parameter set to NO COMMAND by either a warmstart action or TCL command, their tags are also shown with a white background. Note that the loop warm started for a given channel is the loop whose tag is shown in the S800 I/O Display. If two or more loops reference the same channel, only the last loop downloaded participates in warmstart and is listed on the S800 I/O Display.

## S100 I/O Displays

These S100 runtime displays relate to the Series 100 process interface as described in the *S100 I/O User's Guide* where you can find additional information about S100 I/O.

### S100 LAN Display

The S100 LAN display, [Figure 71](#), shows the submodule status and information about each S100 device configured under a particular LAN. Display links for each device are located in the first five columns of the row for the device. If a redundant device is configured, the display link for the redundant device is activated in the last two columns for the device row. Select the device display link to call up the S100 I/O display for that device.



The screenshot shows a window titled "AC1 : Diag AC410 CPS Status". Below the title bar, there are navigation icons and a dropdown menu showing "AC1:Diag AC410 CPS Status". Below that, the following information is displayed: "Devsub 1200 State ACTIVE Name AC1" and "Lan 1 Mod/Sub 1/5". The main content is a table with the following data:

Device	Templet	Dev Type	Comm Addr	Dev State	Red Dev	Red State
1	AC1_S_B1	DSAX 110	C0	ACTIVE	01 04	LINK NOT WORK
2	AC1_S_B2	DSDX 180	C1	ACTIVE	00 00	NOT CONFIG
3	AC1_S_B3	DSDP 150	C2	ACTIVE	00 00	NOT CONFIG
4	AC1_S_B1	DSAX 110	C3	LINK NOT WORK	01 01	ACTIVE
5	AC1_S_B5	DSDX 180	C4	LINK NOT WORK	00 00	NOT CONFIG

Figure 71. S100 LAN Display

The heading fields on the S100 LAN display are described below.

<b>Devsub</b>	The controller DCN address where the S100 I/O is located.
<b>State</b>	The runtime status of the controller. <b>ACTIVE</b> indicates the submodule is in use. <b>FAILED</b> means the submodule has stopped.
<b>Name</b>	The controller part of the logical controller name created during database configuration is given in this field.
<b>Lan</b>	This field represents the number (1 to 4) of the local area network for all the primary I/O boards listed below.
<b>Mod/</b>	For the AC 460 controller, this field represents the position of the SC510 in the Futurebus+ backplane (3, 4, 7, 8, 11, 12). For the AC 410 controller, it represents the position of PM150 in the 410 controller. This is always = 1.
<b>Sub</b>	For the AC 460 controller this field corresponds to where the S100 interface is attached to the I/O submodule (CI540) on the front of the SC510 card (1 = top, 2 = bottom). For the AC 410 controller it corresponding to the 5th interface (S100 integrated backplane) on the 410 controller.

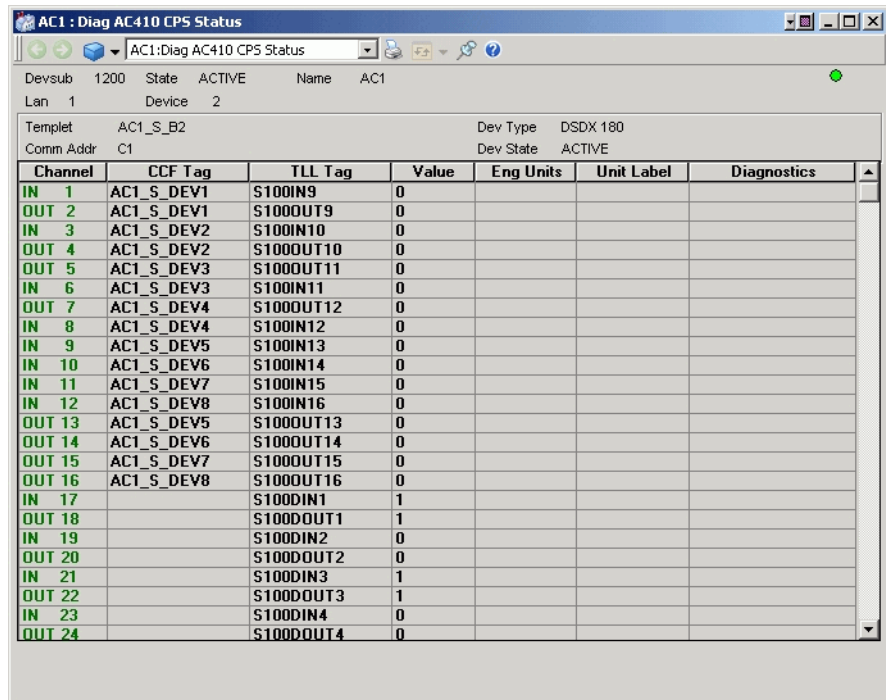
The column headings for each device row on the S100 I/O overview display are described below.

<b>Device</b>	This field is the logical block number of the S100 device as specified during database configuration.
<b>Templet</b>	The logical device name created during database configuration is given in this field.
<b>Dev Type</b>	This field shows the specific I/O board type.
<b>Comm Addr</b>	This field is the hexadecimal communication address assigned during database configuration.

- Dev State** This field shows the runtime status of the I/O board. ACTIVE indicates the primary I/O board is in use, processing I/O signals. FAILED means the primary I/O board has stopped handling I/O signals and the redundant I/O board may be ACTIVE if configured. UNKNOWN indicates not installed.
- Red Dev** If the I/O board has a redundant partner configured, the address of the redundant board is given. The format (SS LL) indicates the submodule and lan of the redundant I/O board. If a redundant board is not configured, '00 00' is displayed.
- Red State** The status of the redundant I/O board. 'Ready' means the redundant I/O board is available to backup the primary I/O board. ACTIVE indicates the redundant I/O board is in use, backing up the primary I/O board. If a redundant I/O board is not configured, this field will show NOT CONFIG. The DSAX 110 can show that both the primary and backup boards are ACTIVE since both boards drive the outputs at 50% of required value.

## S100 Device Status Display

The S100 Device Status Display, [Figure 72](#), shows information for the device and each channel configured for the S100 I/O board (device diagnostics are not displayed). Use the context menu to move up to the LAN display.



The screenshot shows a window titled "AC1 : Diag AC410 CPS Status". The window displays the following information:

- Devs: 1200, State: ACTIVE, Name: AC1
- Lan: 1, Device: 2
- Templet: AC1\_S\_B2, Dev Type: DSDX 180
- Comm Addr: C1, Dev State: ACTIVE

Channel	CCF Tag	TLL Tag	Value	Eng Units	Unit Label	Diagnostics
IN 1	AC1_S_DEV1	S100IN9	0			
OUT 2	AC1_S_DEV1	S100OUT9	0			
IN 3	AC1_S_DEV2	S100IN10	0			
OUT 4	AC1_S_DEV2	S100OUT10	0			
OUT 5	AC1_S_DEV3	S100OUT11	0			
IN 6	AC1_S_DEV3	S100IN11	0			
OUT 7	AC1_S_DEV4	S100OUT12	0			
IN 8	AC1_S_DEV4	S100IN12	0			
IN 9	AC1_S_DEV5	S100IN13	0			
IN 10	AC1_S_DEV6	S100IN14	0			
IN 11	AC1_S_DEV7	S100IN15	0			
IN 12	AC1_S_DEV8	S100IN16	0			
OUT 13	AC1_S_DEV5	S100OUT13	0			
OUT 14	AC1_S_DEV6	S100OUT14	0			
OUT 15	AC1_S_DEV7	S100OUT15	0			
OUT 16	AC1_S_DEV8	S100OUT16	0			
IN 17		S100DIN1	1			
OUT 18		S100DOUT1	1			
IN 19		S100DIN2	0			
OUT 20		S100DOUT2	0			
IN 21		S100DIN3	1			
OUT 22		S100DOUT3	1			
IN 23		S100DIN4	0			
OUT 24		S100DOUT4	0			

*Figure 72. S100 I/O Device Status Display*

The following is a description of the fields on the S100 I/O Device display. See the S100 overview heading descriptions for Devsub, State, Name and Lan. See the S100 overview column heading descriptions for Device, Templet, Dev Type, Comm Addr, and Dev State.

<b>Channel</b>	The input (IN) or output (OUT) and channel number of the device. If the value is green, the Data Quality is good. If the value is red, the Data Quality is bad.
<b>CCF Tag</b> and <b>TLL Tag</b>	The TAG is the name of the CCF loop or the TLL device that uses the channel. If no loop or device is assigned to the channel then the field is blank. Select the CCF or TLL Tag target area to get the appropriate MOD Faceplate. If a tag is configured for warmstart and is warm started, it is shown with a white background.
<b>Value</b>	The actual value of the input from the process or the output value sent to the process. For an output field, you can change the value of an output channel by selecting it and entering a value.
<b>Diagnostics</b>	Not used. See the S100 I/O instruction for error message information.

## Smoothstart Start-up Sequence

Upon acquisition of an S100 I/O device, a diagnostic message identifying the device will be issued to the user. Use the MOD Faceplate or Loop FCM Display to see the following conditions:

- for an analog I/O device, whether the CCF tag which references a channel has an Output FCM in the Manual mode.
- for a digital I/O device, whether the CCF device loop which references a channel has NO COMMAND as its DEV\_CMND attribute.

Upon acquisition the user must:

1. Acknowledge a device acquisition diagnostic message.
2. Call up the corresponding MOD Faceplate or Loop FCM Display to find out what loops are affected.
3. Make any necessary modifications to synchronize the control system with the current outputs.

4. Commission any affected CCF output FCMs by setting their output modes to AUTO.
5. Commission any affected CCF Device loops by issuing device commands as required.

## S100 Warmstart

### Actions Upon Node Startup

Upon node startup, outputs to all devices are disabled. The database is downloaded with each CCF loop output FCM in its initial output mode and each device loop holding its initial command. A period of time is allowed for each S100 board to log into the system. As each board is recognized, its outputs are read. If configured for warmstart, the corresponding CCF loop output FCM for each output channel on the board is placed in the MANUAL mode and its result is synchronized with the field output value in the board memory. The action taken by a device loop configured for warmstart depends on whether there are inputs to the device. If there are inputs to the device, the DEV\_CMND parameter is adjusted to command the device to assume the existing field state. If there are no inputs, the DEV\_CMND parameter is set to the INIT\_CMND (NO COMMAND). Finally, the automatic transfer of outputs to the S100 board is enabled.

### Actions On Board Communication

As a board logs out, a diagnostic message “Block nnnn has logged out” is sent and the associated data quality goes BAD. As a result, FCMs may take their configured “action on output error” and the automatic transfer of outputs to the board is disabled. As a board logs in, output values are read from the board. If configured for warmstart, the corresponding CCF output FCM or device loop for each channel on the board is warm started. That is, the output FCM is placed in the MANUAL mode, and the output value from the board is written to its result attribute. For the warm started device loop, its DEV\_CMND attribute is set to NO COMMAND. Finally, the automatic transfer of outputs to the S100 board is enabled.

### **Actions On Loop Download**

When a loop with an output FCM configured for warmstart is downloaded, the output mode of the FCM is placed in the MANUAL mode and the output value from the board is written to its result attribute. When a device loop configured for warmstart is downloaded, its DEV\_CMND attribute is set to NO COMMAND.

### **Runtime Display Support**

During runtime, but not at startup, as boards log in, diagnostic messages are sent to the diagnostic archive. On the S100 I/O Display, there is a display target for a CCF tag associated with each channel. On this display, the tags for CCF loops with output FCMs and device loops configured for warmstart that were warm started are shown with a white background. Note that the loop warm started for a given channel is the loop whose tag is shown in the S100 I/O Display. If two or more loops reference the same channel, only the last loop downloaded participates in warmstart and is listed on the S100 I/O Display.

### **Operation Action**

With warmstart configured for output FCMs and device loops, the authorized user should:

- acknowledge device acquisition diagnostics (at node startup, it is assumed all boards present are warm started)
- call up the corresponding S100 I/O Display to determine which tags are affected by warmstart
- make any necessary modifications to synchronize the control system with the current outputs
- commission affected CCF output FCMs by setting their modes to AUTO
- commission affected CCF device loops by issuing device commands (if desired)





For continuous loops with Analog and Digital Output FCMs or device loops configured for warmstart that have been warm started, the tag is shown with a white background.

When CCF loops with output FCMs configured for warmstart are placed in the MANUAL mode by either a warmstart or user action, their tags are shown with a white background. When device loops configured for warmstart have their DEV\_CMND parameter set to NO COMMAND by either a warmstart action or TCL command, their tags are also shown with a white background. Note that the loop warm started for a given channel is the loop whose tag is shown in the S100 I/O Display. If two or more loops reference the same channel, only the last loop downloaded participates in warmstart and is listed on the S100 I/O Display.

## PROFIBUS Displays

These PROFIBUS runtime displays relate to a PROFIBUS process interface.

### PROFIBUS LAN Display

The PROFIBUS LAN display, [Figure 73](#), shows the CI541 submodule status (CTL) and information about each device (DEV) configured under a particular LAN. From this display you can:

- Select a device display target (DEV) to call up the PROFIBUS Device display.
- Scroll display as necessary to view all devices (up to 126).
- View the CI541 (CTL) status and its Diagnostic Display.
- Switch the display to another LAN # by returning to the AC460 display.

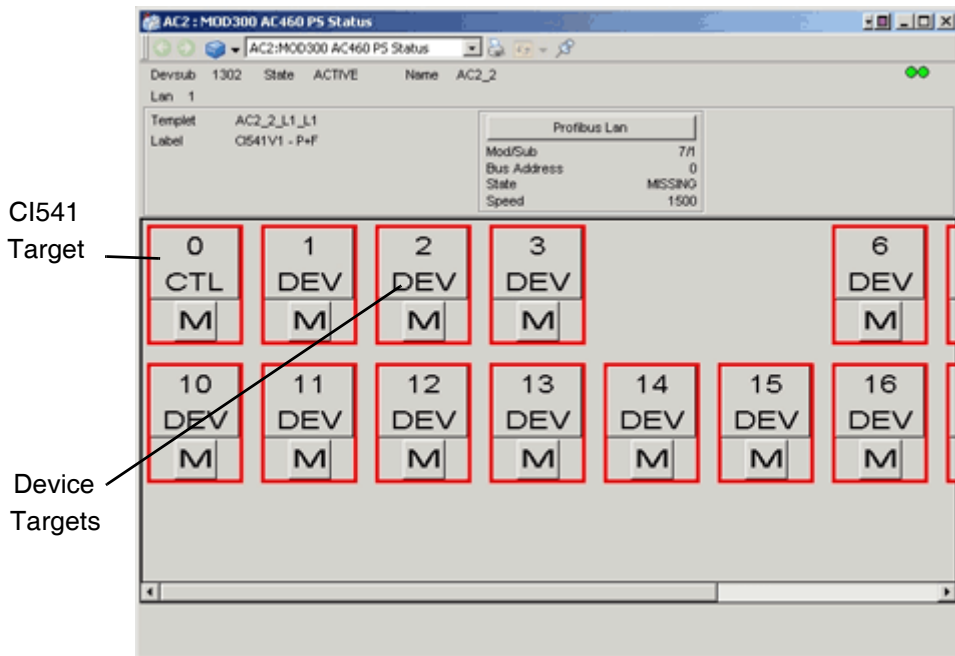


Figure 73. PROFIBUS LAN (Overview) Display

**PROFIBUS LAN Header Area**

The heading fields on the PROFIBUS LAN display are described below.

<b>Devsub</b>	The controller DCN address where the PROFIBUS I/O is located.
<b>State</b>	This field shows the status of the controller as ACTIVE or DOWN.
<b>Name</b>	The controller subsystem part of the logical name created during database configuration.
<b>Lan</b>	This field represents the number of the Local Area Network (LAN) for all the stations shown. For the AC 460 this is LAN # 1 - 4. Go back to the controller display to select a different LAN.
<b>Templet</b>	The logical device name created during database configuration is given in this field.
<b>Label</b>	Descriptive label associated with templet.

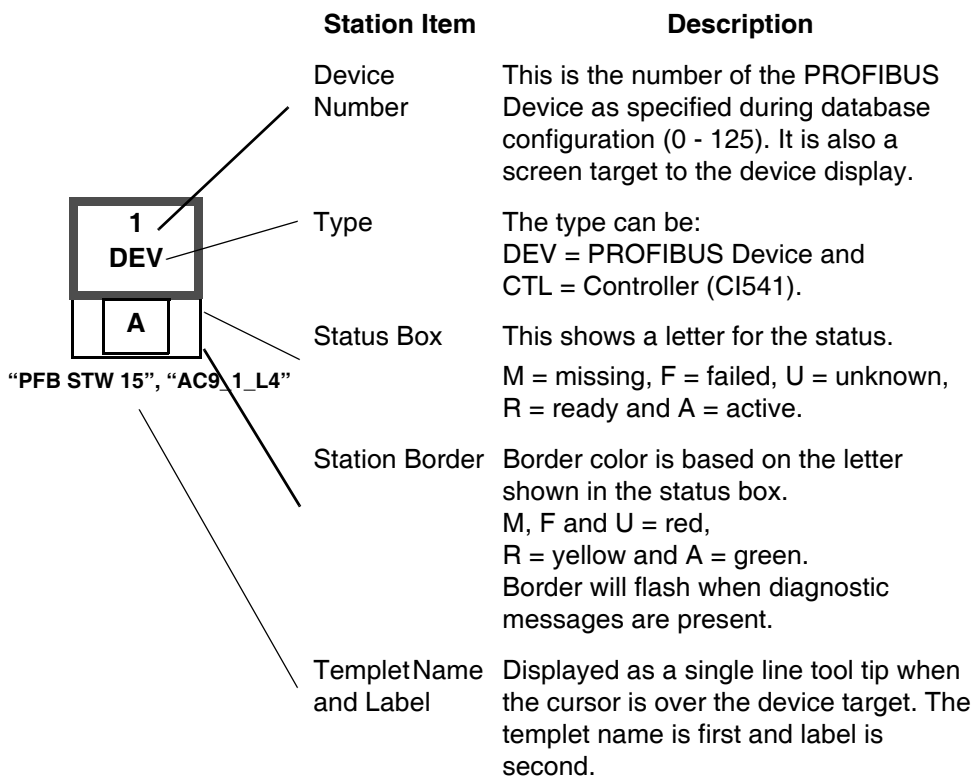
**PROFIBUS LAN Status Area**

The PROFIBUS LAN status area represents the CI541 Interface as described below.

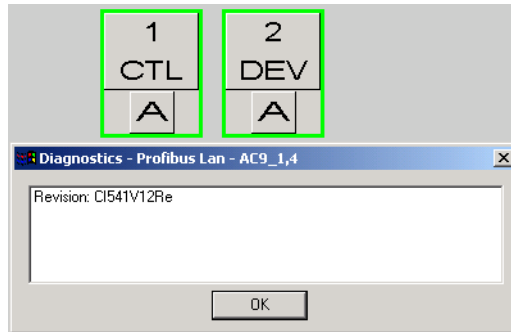
<b>Mod/Sub</b>	This field represents the slot number of the of the controller carrier MODule, and the SUBmodule location of the CI541.
<b>Bus Address</b>	This is the field bus address assigned to the CI541.
<b>State</b>	This field shows the status of the CI541 submodule. ACTIVE indicates the submodule is in use. FAILED means the submodule has stopped. MISSING means there is not a submodule installed in this location.
<b>Speed</b>	This shows the PROFIBUS network Bus Speed (Kbit/sec) as entered on the PRFI_LAN templet.

### PROFIBUS LAN Device Detail

In the main section of the display, a detail window shows specific information about each PROFIBUS Device. The right scroll bar is used to view other devices on the LAN. The display is arranged in a 10 by 8 grid for all devices (0 -125). Devices not configured are not shown. Device 125, being in the lower right corner, can be seen by scrolling the window to that area of the grid. For each device, information is displayed as described below.



If the device is the CI541 type CTL, selecting that icon will bring up the CI541 Diagnostics Display, [Figure 74](#). Currently, there are no LAN diagnostics. Only the revision will be displayed.



*Figure 74. CI541 Diagnostics Display*

## PROFIBUS Device Display

The PROFIBUS Device display is started by selecting a configured device from the PROFIBUS LAN Display. The PROFIBUS Device display, [Figure 75](#), shows information about the modules for the selected device on the current LAN. Select a display target to get a Module/Channel Display. The right scroll bar is used to view other modules. Use the context menu to move up to the LAN display. From this display you can:

- Select a module display target (MOD) to call up the Module/Channel display.
- Scroll display as necessary to view all modules (up to 64).
- View the device diagnostics and device info.
- Call up the display to another LAN # by returning to the AC460 display or navigate back to the LAN display through the context menu.

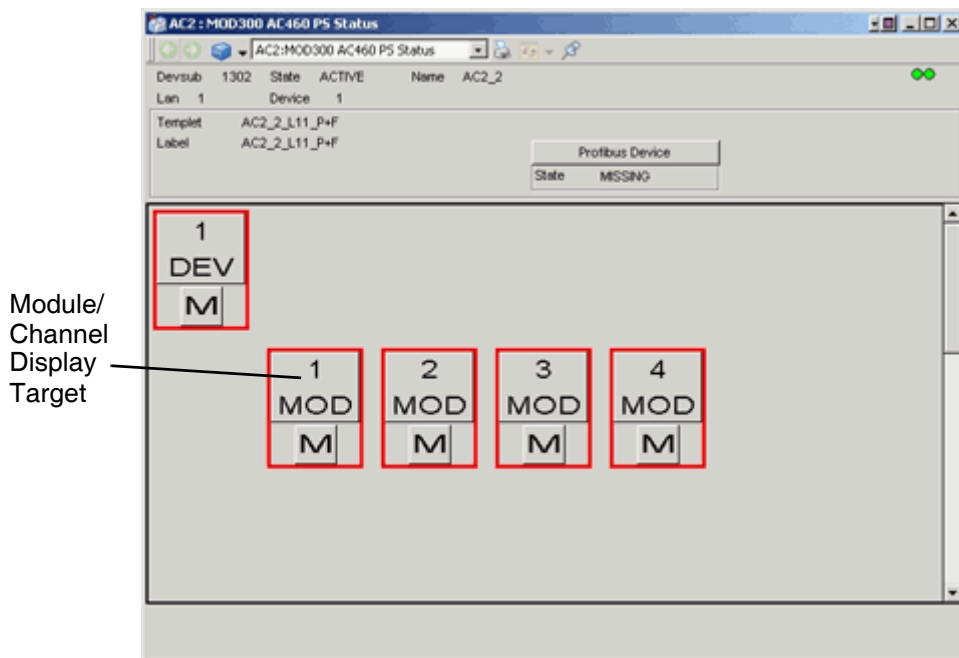
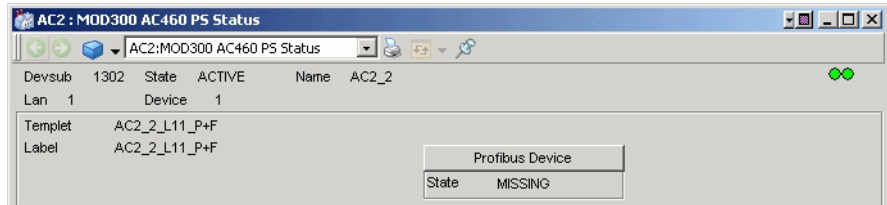


Figure 75. PROFIBUS Device Display

### PROFIBUS Device Header Area

The top of the PROFIBUS Device display identifies the device, the associated AC460 controller and has other information as described below.



<b>Devsub</b>	The controller DCN address where the device is located.
<b>State</b>	This field shows the status of the controller as ACTIVE or DOWN.
<b>Name</b>	The controller subsystem part of the logical name created during database configuration.
<b>Lan</b>	This field represents the number of the Local Area Network (LAN) for all the stations shown. For the AC 460 this is LAN # 1 - 4. Go back to the controller display to select a different LAN.
<b>Device</b>	This field represents the device number. Go back to the LAN display to select a different device.
<b>Templet</b>	The logical device name created during database configuration is given in this field.
<b>Label</b>	Descriptive label associated with templet.

**PROFIBUS Device Status Area**

The PROFIBUS Device area shows the device state and contains a button to display device diagnostics and information. Select the PROFIBUS Device box to select the **Get Diagnostics** or **Get Device Info** display, [Figure 76](#).

- State** This field shows the status of the device.
  - ACTIVE indicates the device is in use.
  - READY indicates a transient state on the way to becoming ACTIVE.
  - FAILED means the device has stopped.
  - MISSING means there is not an device connected.
  - UNKNOWN is an indeterminate state, sometimes seen as a transient state as the station changes state.
  
- Get Diagnostics** Select this to get vendor specific device-related diagnostic codes, [Figure 76](#). You may also get vendor specific device diagnostics by selecting the device detail. See the vendor’s documentation to interpret device-related diagnostic hexadecimal codes.
  
- Get Device Info** Select this to get device information, [Figure 76](#).

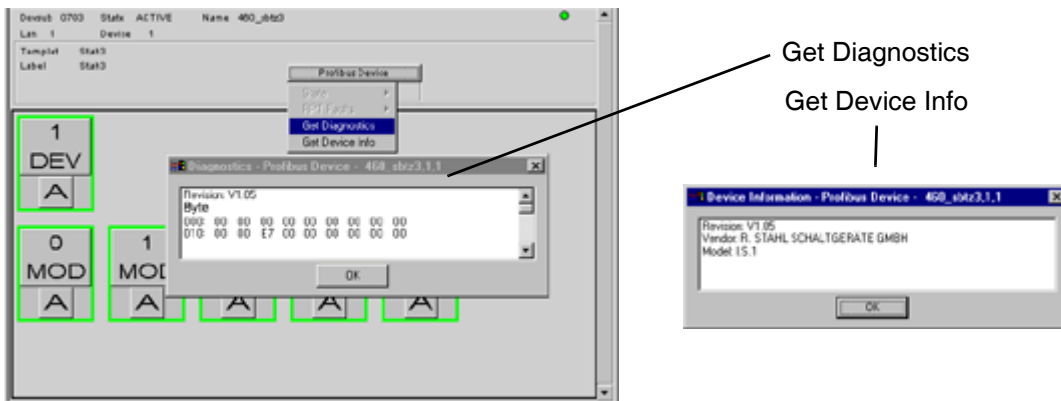


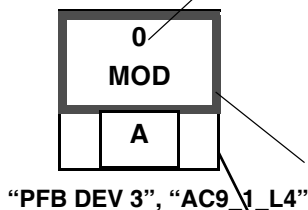
Figure 76. PROFIBUS Device Get Diagnostics and Get Device Info displays



## Device Detail

In the main part of the display, [Figure 75](#), a detail window shows specific information about the devices and each associated I/O module. The right scroll bar is used to view other modules. The information is displayed as described below (see [Figure 75](#) for device graphic).

Station Item	Description
Module Number or Position	This is the position of the I/O module. It is also a screen target to the Module Display. I/O modules not configured are not shown and a blank space is left.
Component Type	The component type is shown under the position number. DEV = PROFIBUS Device MOD = I/O Module.
Status Box	This shows a letter for the module status: M = missing, F = failed, U = unknown, R = ready, and A = active.
Device Border	Border color is based on the letter shown in the status box. M, F and U = red, R = yellow, A = green. Border will flash when diagnostic messages are present.
Templet Name and Label	Displayed as a single line tool tip when the cursor is over the station target. The templet name is first and label is second.



## Module/Channel Display

The PROFIBUS Module/Channel display is started by selecting a configured module from the Device Display. The Module/Channel display, [Figure 77](#), shows information for a single module, including channel type and value, CCF or TLL tag association, channel value, data quality and diagnostics. A module may have a mixture of data and I/O types. Use the context menu to move up to the device or LAN displays. From this display you can:

- Change channel output value
- Select loop CCF tag and start loop faceplate
- Select TLL point tag and start TLL faceplate
- View the module/channel diagnostics.
- Switch the display to another LAN # by returning to the AC460 display.
- Navigate to current LAN or Device through the context menu.

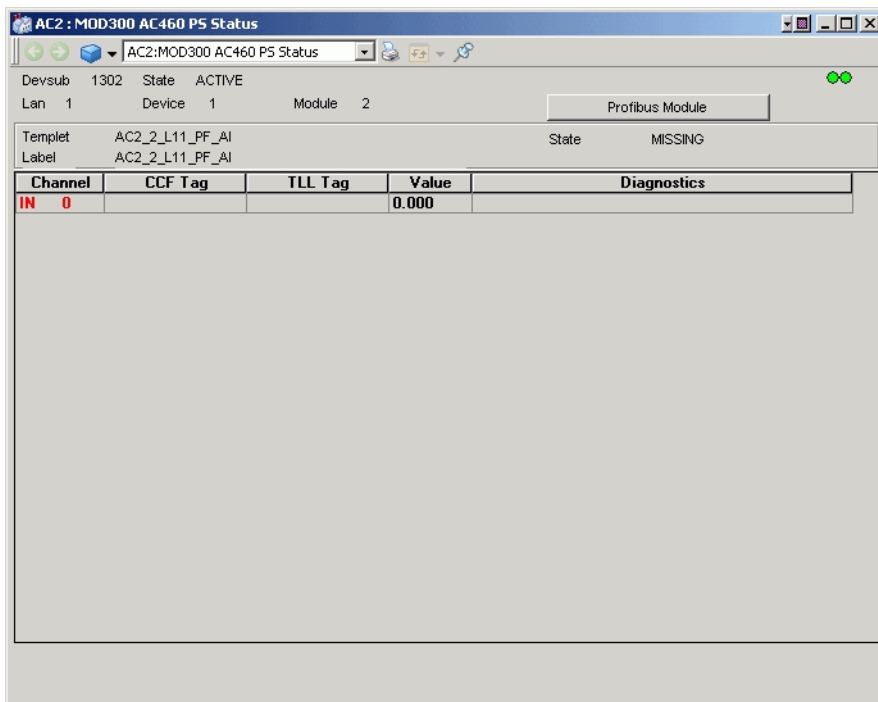
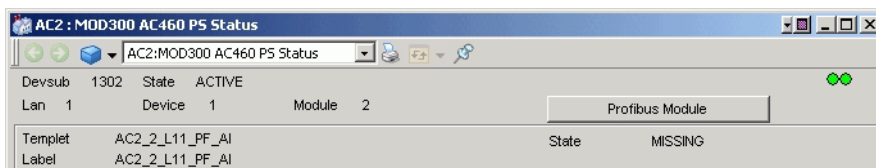


Figure 77. Module/Channel Display

### PROFIBUS Module/Channel Header Area

At the top of the display there is a section related to the addressing of the I/O module as described below.



#### Module Address Info

#### Description

<b>Devs</b>	The controller DCN address where the device is located.
<b>State</b>	This field shows the status of the controller as ACTIVE or DOWN.
<b>Lan</b>	This field represents the number of the Local Area Network (LAN) for the AC 460 device shown (can be 1 to 4). Go back to the controller display to select a different LAN.
<b>Device</b>	This field represents the device number. Go back to the LAN display to select a different device.
<b>Module</b>	This field represents the module number. Go back to the device display to select a different module.
<b>Templet</b>	The logical device name created during database configuration is given in this field.
<b>Label</b>	Descriptive label associated with templet.
<b>State (Module)</b>	This field shows the status of the module. ACTIVE indicates the module is OK. FAILED means the module has stopped. MISSING means there is no module installed.

**PROFIBUS Module/Channel Status Area**

The channel status window section shows specific information about each channel of the module. A scroll bar is used to view other channels of the device. For each channel, several columns of information are displayed as described below.

<b>Channel</b>	The input (IN) or output (OUT) and channel number of the device. The text color for each channel is based upon the data quality: Dark Green = data quality is good. Red = data quality is bad
<b>CCF Tag</b> and <b>TLL Tag</b>	The TAG is the name of the CCF loop or the TLL device that uses the channel. If no loop or device is assigned to the channel then the field is blank. Select the CCF or TLL Tag target area to get the appropriate MOD Faceplate. Select the context menu for CCF or TLL tag to navigate to the Loop FCM or Loop Detail display.
<b>Value</b>	The actual value of the input from the process or the output value sent to the process. For an output field, you can change the value of an output channel by selecting it and entering a value.
<b>Diagnostics</b>	Diagnostics from the PROFIBUS device. See the vendor manual for detailed information.

**Startup**

All output values are set to zero (0) on PROFIBUS startup.

## TRIO Displays

The TRIO runtime displays using MOD 300 are described here. For a complete description of the aspects of TRIO which operators and engineers must understand to perform their runtime functions, refer to the *Taylor Remote I/O (TRIO) User's Guide*. The types of Remote I/O Displays are:

- TRIO (Remote I/O) LAN Display which shows the status of a Field Bus and its attached blocks (up to 30).
- TRIO (Remote I/O) Block Display which gives the status, I/O values and diagnostic messages for one of the TRIO blocks.

### Accessing the Remote I/O Displays for a SC Controller

To access the Remote I/O Displays for a SC Controller:

1. Select the desired SC Controller Subsystem and access the Controller Subsystem Status Display.
2. To use the Display Request button, select the desired SC Controller, select the **Display Request** button and then select the desired TRIO LAN to access the TRIO LAN Display.
3. Alternately, you can select a specific Remote I/O in the Diagnostic area to get the associated TRIO LAN display.

### Accessing the Remote I/O Displays for an AC Controller

To access the Remote I/O Displays for an AC Controller:

1. Select the desired AC Controller Subsystem and access the AC460 or AC410 Subsystem Status Display.
2. Select the desired TRIO LAN submodule to access the TRIO LAN Display.

## TRIO LAN Display

The TRIO LAN Display, [Figure 78](#) for AC Controller and [Figure 79](#) for SC Controller, show the status of the TRIO Field Bus and its attached blocks.

From this display you can:

- Activate, Deactivate, Restart or Switch Bus Redundancy (AC Controller)
- Establish, Terminate, or Switch Bus (SC Controller)
- Enable, Disable or Clear All Diagnostics
- Activate all blocks

AC3 : Diag AC460 P5 Status

Devsub 1401 State ACTIVE Name AC3\_1

Lan 1

Redundancy AVAILABLE

Diag. Checking Enabled

Bus A		Bus B	
Mod/Sub	3/1	Mod/Sub	4/2
State	ACTIVE	State	ACTIVE
MSEC	16	MSEC	21
HHM	NOT IN USE	HHM	NOT IN USE

Block	Templet	Block Type	Block Status
1B	AC3_1-T-1	CS ANALOG 6 INP	ACTIVE
2B	AC3_1-T-2	CUR SRC ANALOG	ACTIVE
3B	AC3_1-T-3	CUR SRC ANALOG	ACTIVE
4B	AC3_1-TB4	8 CKT GROUPE	ACTIVE
5B	AC3_1-TB5	32 CKT I/O	ACTIVE
6A	AC3_1-T-6	CUR SRC ANALOG	ACTIVE
7B	AC3_1-TB7	4IN2OUT ANALOG	ACTIVE
8A	AC3_1-TB8	4IN2OUT ANALOG	ACTIVE
9A	AC3_1-TB9	16 CKT OUTPUT	ACTIVE
10B	AC3_1-TB10	RTD INPUT	ACTIVE
11B	AC3_1-TB11	HI SPD COUNTER	ACTIVE
12?	AC3_1-TB12	CS ANALOG 6 OUT	FAILED
13A	AC3_1-TB13	16 CKT I/O	ACTIVE
14A	AC3_1-TB14	8 CKT GROUPE	ACTIVE
15B	AC3_1-T-B15	CUR SRC ANALOG	INIT FAIL
16B	AC3_1-T-16	CUR SRC ANALOG	ACTIVE
17B	AC3_1-T-17	16 CKT I/O	ACTIVE
18A	AC3_1-T-B18	16 CKT INPUT	ACTIVE
19A	AC3_1-T_19	8 CKT GROUPE	ACTIVE
20B	AC3_1-TB20	16 CKT INPUT	FAILED

Figure 78. TRIO LAN Display (AC)

Block	Templet	Block Type	Block Status
1	SC5_2_L1-B1	THERMOCOUPLE	INIT FAIL
2	SC5_2_L1-B2	16 CKT INPUT	INIT FAIL
6	SC5_2-L1-B6	32 CKT I/O	INIT FAIL
7	SC5_2_L1-B7	CUR SRC ANALOG	INIT FAIL
8	SC5_2-L1-B8	RTD INPUT	INIT FAIL
10	SC5_2_L1-B10	8 CKT GROUPED	INIT FAIL
13	SC5_2_L1-B13	16 CKT I/O	INIT FAIL
14	SC5_2_L1-B14	32 CKT I/O	INIT FAIL
16	SC5_2_L1-B16	16 CKT I/O	INIT FAIL
18	SC5_2-L1-B18	32 CKT I/O	INIT FAIL
19	SC5_2-L1-B19	16 CKT I/O	INIT FAIL
21	SC5_2-L1-B21	CUR SRC ANALOG	INIT FAIL
22	SC5_2_L1-B22	8 CKT GROUPED	INIT FAIL
27	SC5_2-L1-B27	32 CKT I/O	INIT FAIL
28	SC5_2-L1-B28	4IN2OUT ANALOG	INIT FAIL
29	SC5_2-L1-B29	16 CKT I/O	INIT FAIL

Figure 79. TRIO LAN Display (SC)

### TRIO LAN Header Area

The heading fields on the TRIO LAN display are described below.

- Devsub**      The controller DCN address where the TRIO is located.
- State**        This field shows the status of the controller as ACTIVE or DOWN.
- Name**        The controller subsystem part of the logical name created during database configuration.
- Lan**          This field represents the number of the Local Area Network (LAN) for all the stations shown. For the AC 460, this is LAN # 1 - 4. For the SC Controller, this is LAN # 1 - 2.
- Go back to the controller subsystem status display to select a different LAN.

## TRIO LAN Status Area

The TRIO LAN status area represents the Field Bus Interface as described below.

**Redundancy** Possible values are:

**AVAILABLE** - Redundancy is configured and enabled. When it is available, the system switches buses when communications with an active block is lost. You are not allowed to switch single blocks between the buses.

**NOT AVAILABLE** - Redundancy is configured, but not enabled (for the AC 460, one of the CI560 TRIO MIB submodules is either MISSING, FAILED, or READY).

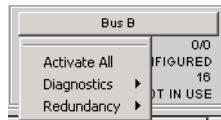
**NOT CONFIGURED** - Redundancy is not configured on the REMOTEIO templet.

**Diag. Checking**

Enabled or Disabled. Use this field to enable or disable diagnostic checking for all blocks on the Field Bus.

**Bus A or Bus B**

Use the context menu associated with the Bus A or Bus B status and select the appropriate command to do the following:



Select **Diagnostics** and from the submenu select:

• **Enable**,

• **Disable** or

• **Clear All** (to clear all diagnostics for every block on the bus)

Select **Activate All** to activate all inactive blocks on the bus (either bus for the AC 460). This is equivalent to all blocks signing into CCF. When a block is not active, CCF channels related to it are given a BAD data quality. This does not change any configuration parameters of the block. If a block was made inactive via its TRIO Block Display, this command can be used to activate it.

Select **Redundancy** and from the submenu select:

- **Establish, Terminate, or Switch Bus** (see [Redundancy on the SC Controller](#) on page 186) or
- **Activate Bus, Deactivate Bus, Restart Bus or Switch Bus** (see [Redundancy on the AC Controller](#) on page 186).



<b>Mod/Sub</b>	This field represents the slot number of the of the controller carrier Module, and the Submodule location of the CI560 TRIO MIB submodule (AC). This field is N/A on the SC controller.
<b>State</b>	<p>This field shows the status of the CI560 submodule (AC).</p> <p>ACTIVE indicates the submodule is in use.</p> <p>FAILED means the submodule has stopped or no redundant bus is configured.</p> <p>MISSING means there is not a submodule installed in this location.</p> <p>READY means the redundant bus is deactivated.</p> <p>This field is N/A on the SC controller</p>
<b>MSEC</b>	This field gives the scan time in milliseconds for the Field Bus as computed by the Advant OCS.
<b>HHM</b>	IN USE, NOT IN USE. This field indicates if a Hand-held Monitor (HHM) is being used on the bus.

**Redundancy on the SC Controller**

You will be asked to Confirm or Cancel redundancy requests.

Use **Establish** on a bus with redundancy configured and NOT AVAILABLE to make it AVAILABLE.

Use **Terminate** on a bus that is AVAILABLE to make it NOT AVAILABLE.

Use **Switch Bus** to force a switchover to the other bus.

**Redundancy on the AC Controller**

You will be asked to Confirm or Cancel redundancy requests.

Use **Deactivate Bus** on a bus with redundancy AVAILABLE, bus state ACTIVE and no blocks using it to change the bus state to READY and redundancy to NOT AVAILABLE.

Use **Activate Bus** on a bus with redundancy NOT AVAILABLE and bus state READY to change the bus state to ACTIVE and redundancy to AVAILABLE.

Use **Restart Bus** if the CI560 TRIO MIB submodule does not come up as expected after a controller reboot or to restart a failed Field Bus Controller.

Use **Switch Bus** to force a switchover of all blocks using the selected ACTIVE bus with redundancy AVAILABLE to the other bus.

### TRIO Blocks on the Field Bus

Each block on the Field Bus is represented by a row on the TRIO LAN Display. To select a TRIO Block Display, move the cursor to the row for the block and click. The block column has either the block number in red or the background in red to indicate the block has an active diagnostic message. The following information is displayed for each block:

**Block** The first cell in a row contains the block number, 1 to 30, for the block on the bus. This number is displayed as follows for the indicated conditions:

Red = diagnostic(s) active

Black = no diagnostics active



On the AC Controller, if this cell has a black background, you can use the context menu to get the **Switch Bus** command (background is red if diagnostic is active). Using Switch Bus here switches only those blocks associated with the BSM (a single stub) providing redundancy is AVAILABLE. You will be asked to Confirm or Cancel the switch bus request.

Also on the AC Controller, the letter A or B after the number indicates which bus the block is using. If the block is FAILED, then a ? replaces the bus letter.

**Templet** This cell contains either the name of the block templet or NOT CONFIGURED if no block templet exists.

**Block Type** The type is displayed here. The actual type may be the same as the corresponding block templet or it may be different.

**Block State** There are four possible block states:

ACTIVE (shown green)

INACTIVE (shown yellow)

FAILED (shown red). The block bus indicator shows a ?.

INIT FAIL - failure to initialize (shown red)

## TRIO Block Display

The TRIO Block Display gives the status, I/O values and diagnostic messages for a selected TRIO block. You access the TRIO Block Display by selecting a Block row on the TRIO LAN Display. [Figure 80](#) and [Figure 81](#) show TRIO Block Displays which represent the appearance and operation of the block displays. The differences among the block displays are in the I/O portions.

From this display you can:

- Return to the subsystem status display to access other TRIO LAN Displays or use the context menu to select **LAN Display**.
- Read or Clear Block Diagnostics
- Activate or Deactivate Block Status
- Open associated CCF or TLL faceplate for a channel (click on CCF Tag or TLL Tag for desired channel)

CHANNEL	CCF TAG	TLL TAG	VALUE	DIAGNOSTICS
IN 1	SC5_1-AI1		0	
IN 2			0	
IN 3			0	
IN 4			0	
OUT 1	SC5_1-AOT1		-1	
OUT 2	SC5_1-AOT2		-1	

Figure 80. TRIO Block Display (Analog)

CHANNEL	CCF TAG	TLL TAG	VALUE	DIAGNOSTICS
<b>COUNTER # 1</b>				
1: STROBE			0	
5: PRELOAD			0	
9: OUTPUT			0	UNKNOWN CIRCUIT FAILURE
17: RESET STR			?	
21: RESET PRE			?	
25: ENAB OUT			?	
ACCUM			0	
FREQ			0	
STROBE 1			0	
<b>COUNTER # 2</b>				
2: STROBE			0	
6: PRELOAD			0	
10: OUTPUT			0	UNKNOWN CIRCUIT FAILURE
18: RESET STR			?	
22: RESET PRE			?	
26: ENAB OUT			?	
ACCUM			0	
FREQ			0	
STROBE 1			0	
<b>COUNTER # 3</b>				
3: STROBE			0	
7: PRELOAD			0	
11: OUTPUT			0	UNKNOWN CIRCUIT FAILURE

Figure 81. TRIO Block Display (Counter)

### TRIO Block Header Area

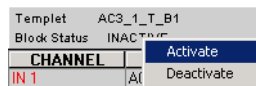
The heading fields on the TRIO Block display are described below.

<b>Devsub</b>	The controller DCN address where the TRIO is located.
<b>State</b>	This field shows the status of the controller as ACTIVE or DOWN.
<b>Lan</b>	This field represents the number of the Local Area Network (LAN) for all the stations shown. For the AC 460, this is LAN # 1 - 4. For the SC Controller, this is LAN # 1 - 2. Go back to the controller display to select a different LAN.
<b>Block</b>	This field identifies the block by its number, 1 to 30. Return to the TRIO LAN Display to select another block.

### TRIO Block Status Area

The TRIO Block status area represents the block as described below.

<b>Templet</b>	This field contains the name assigned to the block templet for this block.
<b>Block Status</b>	There are four possible block states: ACTIVE INACTIVE FAILED INIT FAIL - failure to initialize



Use the context menu on this field to **Activate** (make block status ACTIVE) or **Deactivate** (make block status INACTIVE) the block.

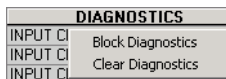
<b>Config Type</b>	This field indicates the type of block templet configured for the block.
<b>Actual Type</b>	This field contains the actual block type.

**TRIO Block I/O Area**

The TRIO Block I/O area represents the I/O data as described below.

<b>Channel</b>	<p>Channel number. The type of channel is indicated as:</p> <ul style="list-style-type: none"><li>IN (Input)</li><li>OUT (Output)</li><li>TRI (Tri state Input)</li><li>BSM (Bus Switch Block controller output)</li></ul> <p>The colors used to display the data quality are:</p> <ul style="list-style-type: none"><li>• green: good data quality</li><li>• red: bad data quality</li><li>• yellow: diagnostic cleared, system is checking to see if problem is corrected</li><li>• flashing color: value forced with HHM, yellow for good data quality, red for bad</li></ul>
<b>CCF Tag</b>	<p>Tag of the loop sending or receiving information from the channel. If multiple tags access the same channel, only one tag is displayed for it.</p> <p>Note that for continuous loops with Analog and Digital Output FCMS or device loops configured for warmstart that have been warmstarted, the tag is shown with a white background.</p>
<b>TLL Tag</b>	<p>A TLL point may also interact with the channel. To view the TLL point, select the TLL Tag cell for the point.</p>
<b>Value</b>	<p>Current value of the channel expressed as one of the following:</p> <ul style="list-style-type: none"><li>0 or 1 for digital</li><li>counts or engineering value for analog</li><li>degrees for thermocouples</li><li>degrees or ohms for RTDs</li><li>A or B for BSM (Bus Switch Module) status</li></ul> <p>Normally, writing an output causes the value to show red until the value is actually updated. However, writing a forced I/O point in manual may stay red because no value is ever written back. Use refresh to update the display.</p>

**Diagnostics** This area displays fault messages that come from the blocks.



The DIAGNOSTICS header is red when a block diagnostic with no associated channel is present. Use the context menu in this column to read **(Block Diagnostics)**

in a Diagnostics window or **Clear Diagnostics**. A single channel diagnostic will appear directly in the related diagnostic cell. You will get a diagnostic window for a channel when the message “Click for multiple channel diagnostics” appears in a row.



When CCF loops with output FCMs configured for warmstart are placed in the MANUAL mode by either a warmstart or operator action, their tags are shown with a white background. When device loops configured for warmstart have their DEV\_CMND parameter set to NO COMMAND by either a warmstart action or TCL command, their tags are also shown with a white background. Note that the loop warmstarted for a given channel is the loop whose tag is shown in the TRIO Block Display. If two or more loops reference the same channel, only the last loop downloaded participates in warmstart and is listed on the TRIO Block Display.



## Direct I/O Displays

Controller I/O (Direct I/O for a Controller) applies only to a SC or Model B Controller. The Controller I/O display shows all configured direct I/O for the selected SC or Model B Controller.

Access the Direct I/O Display, [Figure 82](#), from the Controller Subsystem Status display by selecting an active controller, then selecting the [Display Request](#) button and selecting Controller I/O. Alternately, you can select a direct I/O device in the Diagnostics window of the Controller Subsystem Status display.

The screenshot shows a software window titled "B1\_SUBSYS: Diag MOD Controller Subsystem". Below the title bar, there are navigation icons and a dropdown menu showing "B1\_SUBSYS:Diag MOD Controller Sub". The main area is divided into two sections: "Analog I/O" and "Digital I/O".

**Analog I/O Table:**

Channel	CCF Tag	Value	Unit
IN 1	B1_1-AI1	0.000	V
IN 2		0.000	V
IN 3		0.000	V
IN 4		0.000	V
IN 5		0.000	V
IN 6		0.000	V
IN 7		0.000	V
IN 8		0.000	V
IN 9		0.000	V
IN 10		0.000	V
IN 11		0.000	V
IN 12		0.000	V
IN 13		0.000	V
IN 14		0.000	V
IN 15		0.000	V
IN 16		0.000	V
OUT 1	B1_1-AOT1	12.224	MA
OUT 2	B1_1-AOT2	11.829	MA
OUT 3	B1_1-AOT3	11.435	MA
OUT 4	B1_1-AOT4	11.035	MA
OUT 5	B1_1-AOT5	10.635	MA
OUT 6	B1_1-AOT6	10.235	MA
OUT 7	B1_1-AOT7	9.835	MA
OUT 8	B1_1-AOT8	9.429	MA

**Digital I/O Table:**

Channel	CCF Tag	TLL Tag	Value
IN 1	B1_1-DIN1		0
IN 2			0
IN 3			0
IN 4			0
OUT 5	B1_1-PDO1		0
OUT 6			0
OUT 7			0
OUT 8			0
IN 9			0
IN 10			0
IN 11			0
IN 12			0
OUT 13			0
OUT 14			0
OUT 15			0
OUT 16			0
IN 17			0
IN 18			0
IN 19			0
IN 20			0
OUT 21			0
OUT 22			0
OUT 23			0
OUT 24			0
IN 25			0
IN 26			0

Figure 82. Direct I/O Display

From this display you can:

- Return to the subsystem status display using the controller context menu (right click anywhere except in a valid CCF or TLL tag field)
- Open associated CCF or TLL faceplate for a channel (click on CCF Tag or TLL Tag for desired channel)

CCF can use any I/O type (Analog Input, Analog Output, Pulsed Input, Digital Input, Digital Output and Device Loop).

TLL uses only the Digital I/O. Direct I/O for a controller can supply up to 40 regular digital channels and eight interrupting contact input channels. The number of points provided by direct I/O for a turbo node depends on the number of digital I/O boards in the node.

## Direct I/O Block Header Area

The heading fields on the Controller I/O display are described below.

<b>Devsub</b>	The controller DCN address where the direct I/O is located.
<b>State</b>	This field shows the status of the controller as ACTIVE or DOWN. If the controller goes down, existing data is cleared.
<b>Name</b>	The controller subsystem part of the logical name created during database configuration.

## Direct I/O Area

The Direct I/O area represents the Analog I/O and Digital I/O data as described below.

<b>Channel</b>	Channel number. The type of channel is indicated as: IN (Input) OUT (Output) The colors used to display the data quality are: <ul style="list-style-type: none"> <li>• dark green: good data quality</li> <li>• red: bad data quality</li> </ul>
----------------	---

<b>CCF Tag</b>	<p>Tag of the loop sending or receiving information from the channel. If multiple tags access the same channel, only one tag is displayed for it.</p> <p>Select the CCF Tag target area to get the Loop FCM display (use the context menu for other choices).</p>
<b>TLL Tag</b>	<p>A TLL point may also interact with the channel. If both a CCF tag and a TLL tag are configured, only the CCF Tag will show and the TLL tag will display a ?.</p> <p>When a TLL tag is present (Digital I/O only), select the TLL Tag target area or use the context menu to get the IO Point Faceplate.</p>
<b>Value</b>	<p>Current value of the channel expressed as one of the following:</p> <ul style="list-style-type: none"><li>0 or 1 for digital</li><li>A real number for analog inputs and outputs</li></ul> <p>If the channel type is input, the Value field is not editable. If the channel type is output, the Value field is editable (TRIOCONTROL is used for special access rights as described in <i>800xA for MOD 300 Configuration</i>).</p> <p>Normally, writing an output causes the value to show red until the value is actually updated. However, writing a forced I/O point in manual may stay red because no value is ever written back. Use refresh to update the display.</p>
<b>Unit</b>	<p>For Analog I/O Only. The value in the Unit field is always volts (V) for analog input channels and milliamperes (MA) for analog output channels.</p>



---

## Section 8 Operation Examples

### Typical Operator Activities

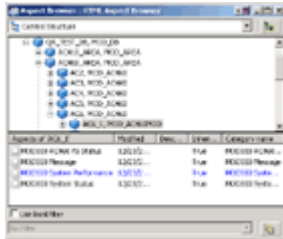
All operator activities related to ongoing operations are performed using the operation displays provided with 800xA for MOD 300. Tuning parameters can be accessed from these displays. Your ability to make tuning adjustment for alarm and control parameters depends on your access rights.

This section describes basic operator activities to illustrate use of the MOD CCF displays in the 800xA System environment. These activities may include:

- Selecting and viewing a control loop
- Monitoring process variable input and setpoint
- Monitoring output to the control device
- Monitoring alarm status
- Acknowledging alarms
- Changing the local setpoint
- Changing operating mode between manual and automatic
- Changing output value while in manual mode
- Changing setpoint source between local setpoint and remote setpoint (if enabled)
- Changing ratio and bias values (if configured)

# Selecting and Viewing a Control Loop

## Object Browser



Structure  
Object Tree  
Aspect List

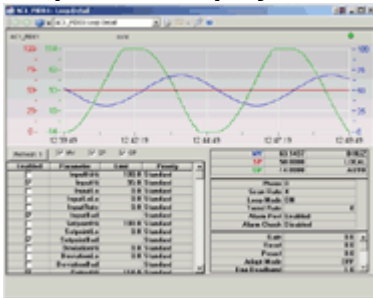
Select an object from the Object Browser using the Control Structure.

Click the object to access the primary display aspect. For a CCF loop, this is the Faceplate.

Right click to get the context menu. This allows you to select either the Loop Detail Display or the Loop FCM Display. The same approach is also used for Diagnostic, TCL and TLL displays.

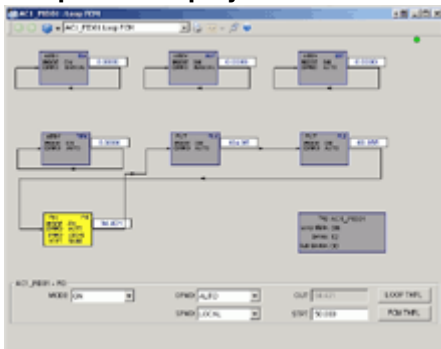
Each object is identified by a user configured tag name.

## Loop Detail Display



Use the Loop Detail Display to enable/disable and change limit parameters, to change loop and tune parameters and to access the associated Loop FCM and Loop Faceplate.

## Loop FCM Display



Use the Loop FCM Display to access the Loop Templet and FCM Templet displays. The Loop FCM Display and its associated templets may be restricted from normal operator use.

**Loop Faceplate (Auto/Local Mode)**

Tag description could indicate that this faceplate applies to PID control loop. The data displayed applies only to the loop which is identified by the tag name.

[Pv] indicates the current measured value.

[Sp] indicates the current setpoint value.

[Out] indicates the current output value.

The current output high limit and low limit values are shown on the Out bar graph scale.

PID loop is operating in automatic with local setpoint [LOC].

To adjust local setpoint from the faceplate, use the setpoint slider or the [Sp] entry field.

**Loop Faceplate (Manual/Local)**

Tag name could indicate that this faceplate applies to a PID control loop.

Current process temperature [Pv] is 31.72 degrees C.

Current value of output in percent is being displayed [Out] 67.99.

Loop is operating in manual with local setpoint [LOC], and Feedback [FB].

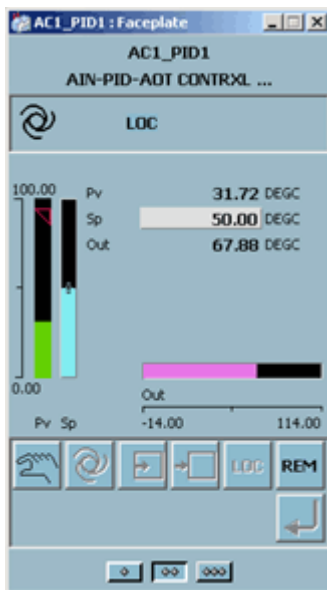
To adjust output in manual, use output slider or the [Out] entry field.

## Single Loop Operation in Automatic Mode

Typical operations for a single control loop in the automatic mode can be with local setpoint, remote setpoint, and remote setpoint with ratio and bias.

### Single Loop Automatic Operation with Local Setpoint

When a control loop is configured for single loop operation with local setpoint, the faceplate displays and control button operations resemble the example below.



indicates that the controller is in automatic mode. Current output value cannot be changed in automatic.



indicates local setpoint is active. Shaded setpoint value field and slide bar on setpoint graphic indicates that the local setpoint can be changed in automatic. Use the slider or the entry value box [**Sp**] to change the value with confirmation.

An engineering units label has been configured for degrees Centigrade (DEGC).



Use the remote button to switch from local to remote.



Use the manual button to switch from automatic to manual. Output can be changed in manual.



## Single Loop Automatic Operation with Remote Setpoint

When a control loop is configured for single loop operation with remote setpoint, the faceplate displays and control operations resemble the example below.



Control loop is in automatic mode.



Remote setpoint is being used.

**[Sp]** indicates the current setpoint value. Remote setpoint cannot be changed locally.

**[Out]** indicates the current output value. Output value cannot be changed in automatic.



Use the local button to switch back to Local from remote [REM]. Setpoint can be changed when in local.



Use the manual button to switch from automatic to manual. Output can be changed in manual.

## Single Loop Automatic Operation with Ratio and Bias

A setpoint bumpless transfer is possible when the balance mode is enabled and the system can manipulate the ratio and/or bias to prevent process bumps. These configurations calculate a ratio or bias value that balances the remote setpoint with the local setpoint upon a switch from local to remote setpoint.

Using auto bias, a switch from local to remote setpoint will cause the bias value to automatically adjust to the difference between the active setpoint and the remote setpoint times the ratio value. For example, if the active setpoint is 170.00 and the remote setpoint is 200.00 with a ratio of 2.00, then the auto bias value will be:

$$-230.00 = 170 - (200 * 2).$$

Using auto ratio, a switch from local to remote setpoint will cause the ratio value to automatically adjust to the ratio between the active setpoint minus bias and the remote setpoint. For example, if the active setpoint is 170.00 and the remote setpoint is 200.00 with a bias of -230.00, then the auto ratio value will be:

$$2 = (170 - (-230)) / 200.$$



**REM**

The loop is in auto with a local setpoint and with local bias.

Adjust the bias value in the **Bi** field to change the active remote setpoint by the local bias value.

The new setpoint appears in the SP field.

Adjust the ratio value in the **Ra** field to change the active remote setpoint by the local ratio value.

The new setpoint appears in the SP field.

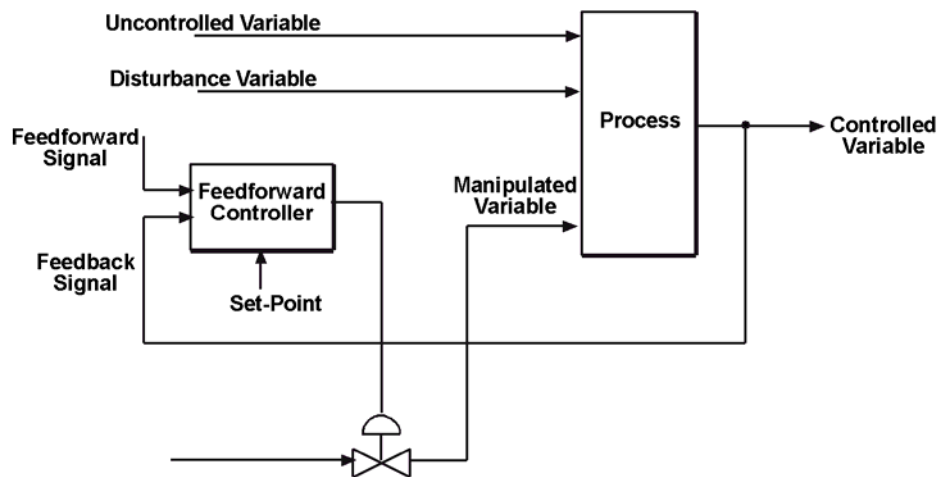
**REM**

Use the remote (REM) button to switch from local to remote setpoint.

Bias value automatically adjusts to balance remote and local setpoints.

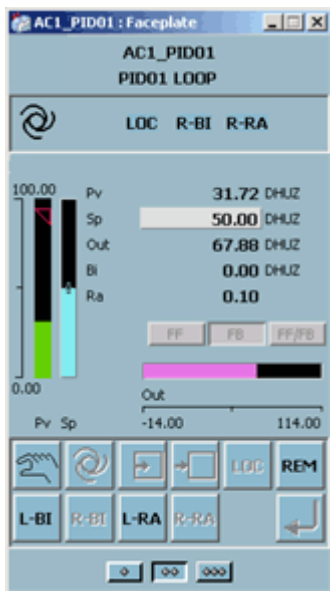
## Single Loop Automatic Operation with Feedforward

Feedforward control is a strategy used to compensate for disturbances in a system before they affect the controlled process variable. A feedforward control system measures a disturbance variable, predicts its effect on the process, and applies corrective action to cancel the effect of the disturbance. A block diagram of this type of system is shown below.



This system provides a combination of feedforward and feedback control. Feedback provides its normal function of holding the process at the setpoint. Feedforward helps the feedback function by modifying the control output as required to cancel the effect of variations in the disturbance variable. The feedforward input (set up as a percent of output) is multiplied by a constant (gain), and a bias value (%) is added to or subtracted from the signal. The gain and bias values are tunable. The feedforward function can be configured to either add the signal to the control output or multiply the output by the signal.

The faceplate display and control button operations for feedforward control resemble the example below.



indicates that the controller is in automatic with feedforward input.



indicates local setpoint.



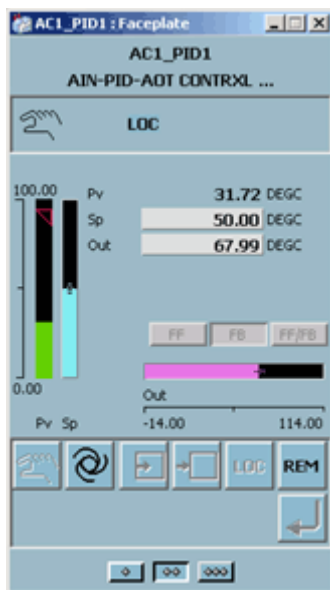
indicates that the loop is in the Feedback mode. Use the [FF] or [FF/FB] buttons to switch to the Feedforward or Feedforward-Feedback modes.

## Single Loop Auto/Manual Transfer

Operations required in transferring a single control loop between the automatic and manual control modes, and the influence of automatic or manual reset on these operations is described here.

### Transfer from Manual to Automatic

The faceplate displays and control button operations for the transfer from manual to automatic resemble the example below.



Indicates that the controller is in manual. Use Out field to adjust output in manual.



indicates local setpoint is active. The local setpoint can be changed in automatic. Use the setpoint slider or the value box [SP] to change the value with confirmation.



Press Auto button to transfer from manual to automatic control. When the controller is in automatic, the output value field is no longer highlighted and the output slider disappears.



Press Remote button to switch the setpoint from local to remote. When the controller is in remote, the setpoint value field is no longer highlighted and the setpoint slider disappears.

Switching from Manual to Auto can be made bumpless by using the techniques that follow:

**Transfers In A Loop With Ratio Or Bias Balancing**

The Balance Mode field on the FCM templet is used to specify either ratio or bias automatic adjustment by the system so the Auto Output Value of the controller is equal to the Manual Output Value.

**Transfers In A Loop With Output Tracking**

The output of a PID FCM supplying the input for the Auto/Manual Controller FCM can be configured to track the TRAK-VAR parameter of the Auto/Manual Controller FCM.

**Transfers In A Loop With Automatic Reset Balancing**

When the control algorithm has reset (integral) response, transfers from automatic to manual and from manual to automatic are always bumpless. After switching from manual to automatic, the process variable is under automatic control in response to the active setpoint, either local or remote.

If the control loop is configured for local setpoint tracking, the local setpoint tracks the process value when the loop is in manual. The process is always at the local setpoint when a transfer to automatic is made, regardless of setpoint limits

If the control loop does not have local setpoint tracking, the local setpoint and the process variable may not be at the same value at the time of a transfer from manual to automatic. The transfer is still bumpless, but immediately after the transfer the process ramps toward the local setpoint value at the reset rate.

**Transfers In A Loop With Manual Reset**

If the control loop does not have automatic reset (integral) response, any setpoint/process offset at the time of a transfer from manual to automatic is maintained. If the loop is configured to provide procedureless manual reset, the manual reset value required to eliminate the offset is automatically calculated while the loop is in manual, and this calculated value is used to remove the offset following a transfer to automatic.

If the loop does not have procedureless manual reset, a setpoint/process offset after transfer to automatic can be eliminated by adjusting the manual reset value.

## Transfer from Automatic to Manual

The faceplate displays and control button operations for the transfer from automatic to manual resemble the example below. During operation, switching from Auto to Manual mode requires one step and is bumpless because the Manual output value automatically tracks the Auto output value when the FCM is in Auto. You can then manipulate the output value in manual.



Indicates controller in automatic.



Indicates local setpoint is active.



Press Manual button to transfer from manual to automatic control. When the controller is in manual, the current output value can be adjusted.



Press Remote button to switch the setpoint from local to remote. When the controller is in remote, the setpoint value field is no longer highlighted and the setpoint slider disappears.

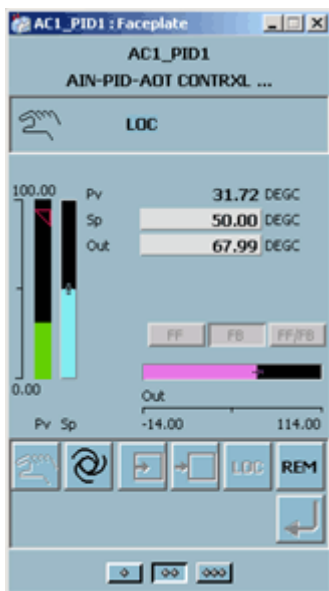


If this indicator is present, the local setpoint is tracking the process value while the loop is in manual so that the setpoint and process are equal when the loop is transferred back to

automatic.

## Single Loop Operation in Manual Mode

This section describes operations when a control loop is in manual. The faceplate displays and control button operations resemble the examples below.



indicates that the controller is in manual. Current output value display [OP] is in percent and can be adjusted using:

- the output slider
- the [OP] entry menu



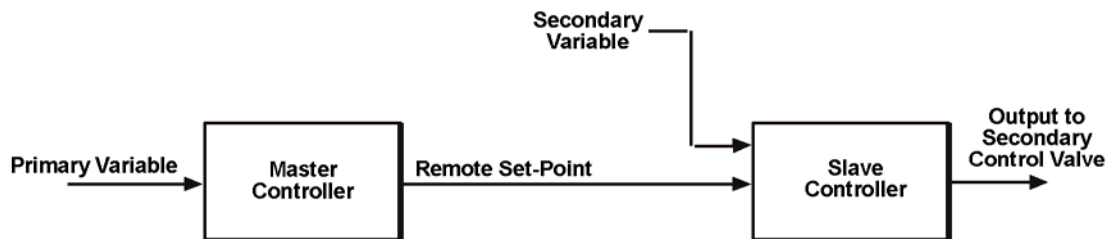
indicates local setpoint is active, and can be changed. Use the slider or the value box [SP] to change the value with confirmation. Since the controller is in manual, the process is following the manual controller output and not the controller algorithm.



## Cascade Operation

A cascade control system is a multiple-loop system where the primary variable (in the master controller) is controlled by adjusting the setpoint of a related secondary variable (in the slave controller). The secondary variable then effects the primary variable through the process. A block diagram of this type of cascade system is shown below.

The main objective in cascade control is to divide an otherwise difficult to control process into two portions, whereby a secondary control loop is formed around a major disturbance, thus leaving only minor disturbances to be controlled by the primary controller.

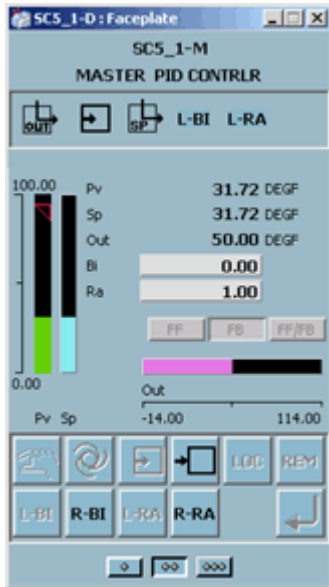


### Cascade Operation in Manual

While in manual, the slave setpoint tracks (automatically remains equal to) the slave process variable. This characteristic is optional by enabling setpoint tracking on the slave. It will eliminate the need for the controller to respond immediately to any deviation which exists when the mode is changed from manual to automatic.

Without setpoint tracking on the slave, the controller responds to the deviation which exists by ramping the output at the reset rate established when the loop was tuned.

**Master Controller**



Slave setpoint tracks slave process variable.

Master output tracks slave setpoint.

Slave output can be adjusted manually.

Master setpoint tracks master process variable.

**Slave Controller**



Meanwhile, the output of the master controller tracks a signal from the slave controller which represents the setpoint of the slave automatically adjusted for ratio or bias. The requirement for balancing the local and remote setpoint signals prior to placing the slave controller in cascade mode is eliminated.

It may also be desirable for the setpoint of the master to track the master process variable. This depends on the desirability of permitting the setpoint of the master to vary indirectly in response to changes at the slave controller.

**Cascade Operation in Auto**

Since the control element (valve, air damper, etc.) is directly manipulated by the slave, the cascade master is configured to permit only auto mode which simplifies operations by ensuring that all mode transfers and manual intervention can be accomplished exclusively at the slave.

## Master Controller



Remote slave setpoint tracks master output.

## Slave Controller



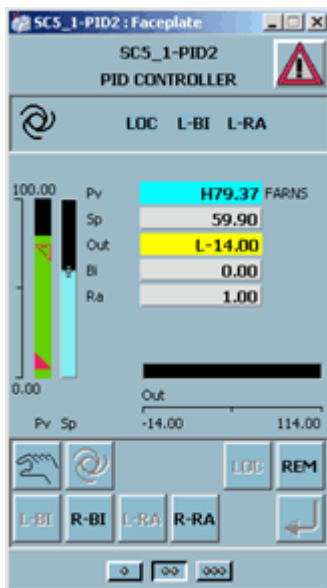
Local setpoint on Master.

- The slave setpoint tracks (automatically remains equal to) the slave process variable while in manual. This feature eliminates the need for the slave to respond immediately to any deviation when the mode is changed from manual to auto (cascade). Without setpoint tracking, the response to any deviation would be to ramp the slave output at the reset rate.
- The output of the master tracks the setpoint of the slave while the slave is in manual. This ensures that the local (slave) setpoint and remote setpoint (master output) are in balance prior to placing the slave loop in cascade (auto) mode.
- The setpoint of the master tracks the master process variable while the slave is in manual. This causes the setpoint of the master to vary indirectly in response to manual output changes in the slave loop. When the system is switched to auto, the master setpoint switches from track to local, and there is no process upset because the track function eliminates any deviation in the master loop.
- Since the control element is directly manipulated by the slave, the master is configured for automatic mode only. This ensures that all mode transfers and manual intervention can be accomplished exclusively at the slave.

## Viewing Alarms on MOD Loop Displays

Alarms can be configured for each control loop to respond to an alarm condition on the process variable, the control output, and the deviation between the setpoint and process value (as a normalized measured value into the PID FCM). Alarm types can be high, high high, low, low low or limit alarms. The parameters of these alarms can be adjusted during operation on the loop detail display.

### CCF Control Loop Faceplate with Alarms



[Pv] indicates that this display applies to the measured variable while the colored background shows an active alarm. The Pv field color blinks if the alarm is unacknowledged.

The Pv bar graph indicates that the alarm is activated by a high input. Alarm trip point on bar graph is indicated by matching colored marker.

[Out] indicates that this display applies to the control output. An output alarm is activated by reaching the output limit. Flashing background indicates that the alarm is active and has not been acknowledged.



Acknowledge all alarms for a tag at once using either the active Alarm Acknowledge button, or by using **Acknowledge** in the context menu.



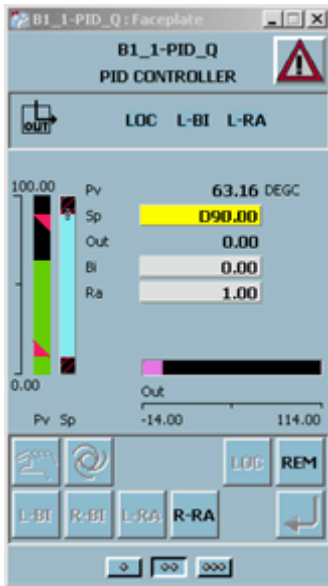
Indicates an active acknowledged alarm (current value is outside the trip point). Field colors stop blinking.



Indicates an unacknowledged alarm that has returned to normal (current value is inside the trip point). Acknowledge the alarm by selecting this button.

Background returns to normal when alarm is inactive and acknowledged.

## Setpoint/Process Deviation Alarm Display



This display is set to alarm on the deviation between the PID setpoint and the normalized measured value input to the PID.

[D] indicates that the alarm is activated by increasing deviation. Flashing background indicates that the alarm is active and has not been acknowledged.



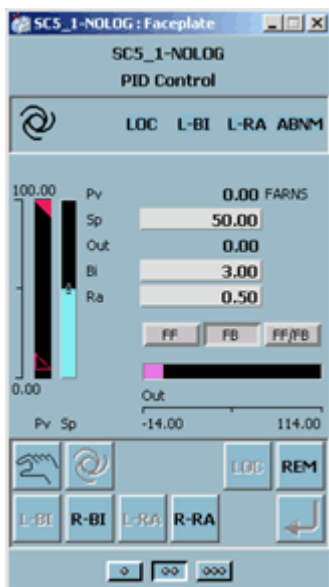
Acknowledge all alarms for a tag at once using either the active Alarm Acknowledge button, or by using **Acknowledge** in the context menu.

Steady colored background indicates an active acknowledged alarm (current value is outside the trip point).

Background returns to normal when alarm is inactive and acknowledged.

## Viewing Abnormal State on MOD Loop Displays

The Abnormal State field specifies whether the system should check the loop for an abnormal condition.



**ABNM**

Indicates an abnormal condition. An abnormal condition for a continuous loop is caused by any one of the following conditions:

- Loop is off
- An FCM is off
- Output of an FCM which is not a PID or AM Controller FCM is in MANUAL
- Output mode of a PID Controller FCM or an AM Controller FCM is not in its design state as defined on its templet
- Setpoint mode of a PID Controller FCM is not in its design state as defined on its templet

When set to YES, the system checks for an abnormal state. When an abnormal state exists, the condition is displayed as ABNM in the status area.

---

# Appendix A Runtime Templates

## General

The runtime versions of the Loop Definition and FCM templates are used to make tuning changes that cannot be made from the faceplate and device loop displays. Their information is always current since it comes directly from the active data base. Some fields are for reference only and are dimmed, other fields are black and are tunable.

Tuning changes are made to database items by first changing the information on the templet and then sending the templet information to the database by selecting the **SAVE** button. If the save process is not successful, a message describing the problem is displayed. After corrective action is taken, you can re-try the save process.

Saving changes both the active database in the subsystem containing the loop and the installed database. However, the original configuration templates are not modified until the database is decompiled.

## Loop Definition Templet

This section describes the active runtime fields on the Loop Definition Templet.

### **LOOP DESCRIPTOR Field**

Specifies a descriptor for the loop. The descriptor is used with the tag to identify the loop. Valid entry is: any combination of up to 24 alphanumeric characters. All 24 characters are printed when information for the loop appears on the Alarm/Event Log, and only the first 12 characters are shown when the descriptor appears on a display.

**PROCESSING RATE Field**

Specifies the interval at which the loop is processed (scanned). This field, the PROCESSING PHASE, and BASERATE fields interact to determine the actual loop processing rate. Scan Rate (Loop Detail display) equals the Processing Rate divided by the Base Rate. If the Base Rate is 0.5 and the Processing Rate is 1.0, then the Scan Rate is 2.0.

Valid entry is in the form of an integer equivalent of hr:min:sec.

The minimum acceptable value is the base rate of the module (0.1 to 1.0) as specified on the CCF Templet. The maximum value is 65535 times the base rate (if the base rate is 0.25 then the maximum is 16383.75). The value entered on this field is automatically rounded up to the next higher multiple of the base rate.

Loop Definition Template entry examples:

<b>Runtime Entry</b>	<b>Configurator Entry</b>	<b>Interpreted as:</b>
0.25	.25	.25 seconds
5	5	5 seconds
257	4:17 or 257	4 minutes 17 seconds
7744	2:09:04 or 7744	2 hours 9 minutes 4 seconds

**PROCESSING PHASE Field**

Use this field to distribute processing for the loops about the processing interval. This field, the PROCESSING RATE, and BASERATE fields interact to determine the loop processing rate. Valid entry is: any integer from 0 to PROCESSING RATE / BASERATE - 1 (rounded up to the next higher integer if not a whole number). If an entry is greater than the valid maximum, the phase is set to 0.

**TREND RATE Field**

Specifies the rate at which the trend data for the measured variable is collected. Valid entries are the following number of seconds:

0 (no trend collection), 6, 12, 60, 120, 360, 720, 1440, 2880



**LO ENG. UNIT LIMIT Field**

Specifies the lower boundary for the trend graphs. Valid entries are: a decimal number.

**HI ENG. UNIT LIMIT Field**

Specifies the upper boundary for the trend graphs. Valid entries are: a decimal number.

**MEASUREMENT UNITS Field**

Specifies engineering units labels. The labels are used when the measured variable of the loop is displayed on operational displays. Valid entry is: any string of up to 6 alphanumeric characters, for example, GPM, PSIG

**LOOP STATE Field**

This field determines whether the loop is processed. When LOOP STATE is OFF, the loop processor ignores the loop. Valid entries are: ON (process the loop), OFF (do not process the loop).

**SUPPRESS ALARMS Field**

This field determines if the loop processor checks that alarms defined by this loop are active. Valid entries are: YES (suppress alarms - do not check), NO (do not suppress alarms).

**ENABLE POSTING Field**

This field specifies whether posting is enabled for the loop. Posting affects the way alarm information is presented. Valid entries are: YES (enable posting), NO (disable posting).

**CUTOUT STATE Field**

This field defines the state the cutout source must be in to cause cutout. Cutout allows posting of the alarms to control the loop from an FCM of this or another loop. Valid entries are:

TRUE	causes cutout when the cutout signal is true (non-zero)
FALSE	causes cutout when the cutout signal is false (zero)

**ENABLE HI ALARMS? Field**

This field specifies whether the high alarm for the measured variable is enabled. Valid entries are: YES (enable the alarm), NO (disable the alarm).

**HI LIMIT Field**

This field defines the high alarm limit for the measured variable. The alarm becomes active when the measured variable becomes equal to or greater than the high limit and clears when the measured variable decreases to the value of the high limit minus the alarm deadband. Valid entry is: a decimal number.

**HIGH ALARM PRIORITY Field**

This field sets the alarm priority which determines how the alarm is displayed. You can specify different colors to distinguish high, medium, and standard priority alarm indicators. Valid entries are: STD (standard priority), MED (medium priority), HIGH (high priority).

**ENABLE LO ALARMS? Field**

This field determines if the low alarm for the measured variable is enabled. Valid entries are: YES (enable the alarm), NO (disable the alarm).

**LO LIMIT Field**

This field defines the low alarm limit for the measured variable. The alarm becomes active when the measured variable becomes equal to or less than the low limit and clears when the measured variable increases to the value of the low limit plus the alarm deadband. Valid entry is: a decimal number.

**LO ALARM PRIORITY Field**

This field specifies the priority level for the low alarm. Valid entries are: STD (standard priority), MED (medium priority), HIGH (high priority).

**BAD MEASURE ALARMS? Field**

This field specifies whether the bad data quality alarm for the measured variable is enabled. The alarm becomes active whenever the data quality of the measured variable becomes BAD. Valid entries are: YES (enable the alarm), NO (disable the alarm).

**BAD ALARM PRIORITY Field**

This field specifies the priority level for the bad data quality alarm. Valid entries are: STD (standard priority), MED (medium priority), HIGH (high priority).

**ENABLE RATE ALARMS? Field**

This field specifies whether the rate alarm for the measured variable is enabled. The rate alarm becomes active when the measured variable changes by an amount greater than the specified rate limit from one loop processing scan to the next. Valid entries are: YES (enable the alarm), NO (disable the alarm).

**RATE LIMIT Field**

This field defines the rate alarm limit for the measured variable. The alarm becomes active when the variable changes at a rate greater than the limit. Valid entry is: a number in engineering units, representing the greatest amount the variable can change per second.

**RATE ALARM PRIORITY Field**

This field specifies the priority level for the rate alarm. Valid entries are: STD (standard priority), MED (medium priority), HIGH (high priority).

**ENGU ALARM DEADBAND Field**

This field defines a deadband for the high, low, high high, and low low alarms on the measured variable. It also becomes the deadband for any setpoint low or setpoint high alarm for the loop. Valid entry is: a floating point number in the engineering units of the loop.

**ENABLE HIHI ALARMS? Field**

This field specifies whether the high high alarm for the measured variable is enabled. This is a second high alarm level for the measured variable. For example, the high alarm for a flow is set at 200 GPM and the high high alarm is set at 230 GPM. When the flow becomes equal to or higher than 200 GPM, the high alarm becomes active. If the flow increases to a rate equal to or larger than 230, both the high high and the high alarms are active. Valid entries are: YES (enable the alarm), NO (disable the alarm).

**HIHI LIMIT Field**

This field defines the high high alarm limit for the measured variable. The alarm becomes active when the measured variable becomes equal to or greater than the high high limit and clears when the variable decreases to the value of the high high limit minus the alarm deadband. Valid entry is: a decimal number.

**HIHI ALARM PRIORITY Field**

This field specifies the priority level for the high high alarm. Valid entries are: STD (standard priority), MED (medium priority), HIGH (high priority).

**ENABLE LOLO ALARMS? Field**

This field specifies whether the low low alarm for the measured variable is enabled. This is a second low alarm level for the measured variable. For example, the low alarm for a flow is set at 100 GPM and the low low alarm is set at 80 GPM. When the flow becomes equal to or less than 100 GPM, the low alarm becomes active. If the flow decreases to a rate equal to or lower than 80, both the low low and the low alarm are active. Valid entries are: YES (enable the alarm), NO (disable the alarm).

**LOLO LIMIT Field**

This field defines the low low alarm limit for the measured variable. The alarm becomes active when the measured variable becomes equal to or less than the low low limit and clears when the measured variable increase to the value of the low low limit plus the alarm deadband. Valid entry is: a decimal number.

**LOLO ALARM PRIORITY Field**

This field specifies the priority level for the low low alarm. Valid entries are: STD (standard priority), MED (medium priority), HIGH (high priority).

**UNIT ID Field**

This field assigns the loop to a unit. This is useful for systems that use the History software. The alarm/event history messages for all loops in a unit are collected and stored in the same file. If a unit assignment is not made for a loop, the History Services software stores alarm/event history messages for the loop in a default unit file. Valid entry is: the name of a unit as defined in the TEMPLLET NAME field on the Unit Master Template for the unit.

**COMP MODE RESTRICT Field**

COMP MODE (computer mode) is used to specify how certain PID or an AM controller loop parameters are handled. During configuration of the Loop Definition Template, you can set loop control to either restricted or not restricted. There are two entries for this field. They are: NO (No restrictions on control of the protected attributes. The computer mode feature is used.), YES (Control of the protected attributes is restricted to either the operator or the computer. Computer control indicates that a TCL program or a program running on a VAX™, IBM PC™ or MODCOMP computer attached to the MOD 300 control system, is manipulating PID or AM parameters.)

**ABNORMAL STATE Field**

This field specifies whether the system should check the loop for an abnormal condition. An abnormal condition for a continuous loop is caused by any one of the following conditions:

- Loop is off
- An FCM is off
- Output of an FCM which is not a PID or AM Controller FCM is in MANUAL
- Output mode of a PID Controller FCM or an AM Controller FCM is not in its design state as defined on its template

- Setpoint mode of a PID Controller FCM is not in its design state as defined on its templet

Valid entries are:

- |     |  |
|-----|--|
| YES | system checks for an abnormal state. If an abnormal state exists, the NORMSTAT attribute is set to 1 and the status area shows ABNM on the operational displays. |
| NO  | system does not check for an abnormal state.   |

### **DMND PROCESSING MODE Field**

This field specifies whether the specified loop(s) are processed when the loop is processed normally or when it is demand scanned, or both. Valid entries are:

- |        |  |
|--------|--|
| NONE   | no extra processing  |
| NORMAL | process loops in the LOOPS TO PROCESS edit window when this loop is normally scanned               |
| DEMAND | process loops in LOOPS TO PROCESS edit window when this loop is demand scanned                     |
| BOTH   | process loops in LOOPS TO PROCESS edit window when this loop is demand scanned or normally scanned |

### **LOOPS TO PROCESS Edit Window**

This edit window lists the loops to be processed in conjunction with the DMND PROCESSING MODE field and the # LOOPS TO PRESCAN field. Valid entries are: up to 5 loop tags. Each line in the edit window can hold one tag. Do not enter the tag of the loop being configured. Up to five levels of nesting are allowed, that is, loops in the edit window demand process other loops, and so on.

## Device Loop Template

This section describes the active runtime fields on the Device Loop Template. Select the Loop FCM Display and then the device block to get the runtime version of the Device Loop Template. Use the fields at the bottom of the Loop FCM Display to make State, Mode and Command changes.

### LOOP DESCRIPTOR Field

This field specifies a loop descriptor. The descriptor is a label included on operational displays and the Alarm/Event Log. This field does not need a unique entry, so more than one loop can have the same descriptor.

Valid entry is:

any combination of up to 24 alphanumeric characters. All 24 characters are printed when information for the loop appears on the Alarm/Event Log, but only the first 12 characters are shown when the descriptor appears on a display.

### PROCESSING RATE Field

This field specifies the loop processing rate. This field and the CCF BASE RATE field interact to determine the loop processing rate. Scan Rate (Loop Detail display) equals the Processing Rate divided by the Base Rate. If the Base Rate is 0.5 and the Processing Rate is 1.0, then the Scan Rate is 2.0.

Valid entry is in the form of an integer equivalent of hr:min:sec. The range is 0.100 second (min.) to 65535 times the base rate (max.). Processing rate must be greater than or equal to the CCF base rate (0.1 to 1.0). This field is automatically rounded up to the next higher multiple of the base rate.

Device Loop Template entry examples:

Runtime Entry	Configurator Entry	Interpreted as:
0.25	.25	.25 seconds
5	5	5 seconds
257	4:17 or 257	4 minutes 17 seconds
7744	2:09:04 or 7744	2 hours 9 minutes 4 seconds

**PROCESSING PHASE Field**

This field allows for even distribution of loop processing about the processing interval. How this field, the PROCESSING PHASE field, and BASE RATE field of the CCF Templet interact to determine the loop processing rate.

Valid entry is:

any integer from 0 to  $(PR/BR-1)$  rounded up to the next higher integer) where PR is the processing rate of the loop and BR is the base rate of the module containing the loop. If an entry greater than  $(PR/BR-1)$  is made, a compiler warning is generated when the data base is compiled and the phase is set to 0.

**INVERTED INPUTS Field**

The INVERTED INPUTS field is used to make modifications to the device handling algorithm by changing the bit patterns of the feedback (input) signals.

Valid entry is a decimal number identifying which bits are inverted.

Example:

If a device type supports the following 3 feedback states,

1 0 1 0

1 0 0 1

0 1 0 1

entering a 4 to the INVERTED INPUTS field would invert bit 3 (the binary equivalent of decimal 4) so the resulting feedback states are:

1 1 1 0

1 1 0 1

0 0 0 1

Similarly, entering a 10 would invert bits 4 and 2 (the binary equivalent of 10).

**INVERTED OUTPUTS Field**

This field is used to make modifications to the device handling algorithm by changing the bit patterns of the output signals.



Valid entry is:

a decimal number identifying which bits are inverted. See the description of the INVERTED INPUTS field for an example of inverting bits.

### **TRANSITION OVERRIDE Field**

A transition timeout occurs when the system issues a command to a device and the device does not go to the proper state within the time specified by the TIMER field. The TRANSITION OVERRIDE field specifies the action applied to the device command (DEV\_CMND) when a transition timeout occurs.

Valid entries are:

NONE	no change
STATE	set to entry in the DEVICE STATE field (DEV_STAT). If STATE is specified, but the current state does not correspond to a valid command, the action taken is the entry in the SAFE COMMAND field, if a command is specified in that field.
SAFE CMD	action taken is the entry in the SAFE COMMAND field
PREVIOUS CMD	set to the last successful previous command
SPECIAL CMD	set to command specified in the SPECIAL COMMAND field

### **LOOP STATE Field**

This field determines whether the loop is processed. When LOOP STATE is OFF, the loop processor ignores the loop. The configuration entry is the initial loop state for the loop.

Valid entries are:

ON	process the loop
OFF	do not process the loop

### **ALARM PRIORITY Field**

This field specifies the priority level for any alarms occurring for the device.

Valid entries are:

HIGH	high (priority 1)
MED	medium (priority 2)
STD	standard (priority 3)

**POSTING ENABLE Field**

This field indicates whether posting is enabled for the loop. Posting affects the way alarm information is presented at the Operator Stations or Multibus-based consoles and on the Alarm/Event Log.

Valid entries are:

YES	enable posting
NO	disable posting

**SUPPRESS ALARMS Field**

This field specifies whether the loop processor checks to see if the alarms defined by this loop are active.

Valid entries are:

YES	suppress alarms (do not check)
NO	do not suppress alarms

**DEVICE OVERRIDE Field**

This field overrides the feedback timer. When set to YES, there are no Timeout Alarms for the device. See the description of the TIMER field for a description of Timeout Alarms.

Valid entries are:

NO_CHECK	override the feedback timer
NORMAL	do not override the feedback timer

**SIMULATION MODE Field**

This field facilitates testing and simulating device situations with TCL or a host computer. When the Simulation Mode is ON, the device loop does not perform inputs and outputs. When it is OFF, inputs and outputs are handled in the normal manner.

This field defines the initial value of the Simulation Mode. During runtime, you can turn the mode ON or OFF via the Loop/FCM Display for the device loop. You can also change the mode by TCL or by programs run on a host computer.

The effect of the Simulation Mode on device processing depends on the state of the DEVICE OVERRIDE field on the Device Loops Template.

Valid entries are:

ON	device simulation mode is on
OFF	device simulation mode is off

**INITIAL MODE Field**

This field determines the output mode for the loop at the start-up.

Valid entries are:

AUTO	loop is under the control of TCL or a supervisory package
MAN	operator determines output

**INITIAL COMMAND Field**

This field specifies the command sent to the device upon start-up and the warmstart action.

Valid entries are:

- any of the valid commands defined by the descriptor set for the device
- no entry (blank field) sends out a 0 when the device loop goes on-line
- NO COMMAND sends out nothing, that is, there is no processing on start-up. This is often used when the device loop interacts with a programmable controller over a PC interface. The entry NO COMMAND is used to insure the device loop does not affect the programmable controller when the device loop

goes on-line.

### **ABNORMAL OVERRIDE Field**

This field specifies the action to take if an abnormal state change timeout occurs. If a device changes state when no command was issued to the device, the abnormal timer starts counting down from the amount of time specified in the ABNORMAL TIMER field. If it reaches 0, the action specified in this field is applied to the device command parameter (DEV\_CMND).

Valid entries are:

NONE	no change to device command, but abnormal state change alarm is generated
STATE	device command (DEV_CMND) is set to value of Device State (DEV_STAT), abnormal state change alarm is generated
SAFE COMMAND	device command (DEV_CMND) is set to entry in the SAFE COMMAND field, abnormal state change alarm is generated
NONE/NA	no change, no abnormal state change alarm
STATE/NA	set to value of Device State (DEV_STAT), no abnormal state change alarm
SAFE/NA	set to entry in the SAFE COMMAND field, no abnormal state change alarm
PREVIOUS CMD	set to last successful previous command
SPECIAL CMD	device command (DEV_CMND) is set to entry in the SPECIAL COMMAND field

### **ABNORMAL TIMER Field**

This field specifies the amount of time the abnormal timer is set to when an abnormal state change occurs. See the description of the ABNORMAL OVERRIDE field. If you want the specified abnormal override actions to occur, you must enter a non-zero entry. Abnormal state change alarms will occur (if the device is so configured), regardless of the entry in this field.

Valid entry is an amount of time in whole seconds.

**SECONDARY TIMER Field**

This field is used when the device loop uses a user-defined TCL device handling algorithm. It sets the initial time for a countdown timer used by the TCL program which sets up the user-defined device algorithm.

Valid entry is an amount of time in whole seconds.

**TIMER Field**

This field sets the time for the device loop's feedback timer. When the loop sends a command to the device, the feedback timer starts counting down. The loop starts checking its input to see if the device has gone to the desired state. A timeout alarm is generated if the device has not entered the desired state by the time the timer expires.

Valid entry is an amount of time in whole seconds.

**DEV INTERLOCK ENABLE Field**

This field specifies whether the device interlock is enabled or disabled. When it is enabled, the result of the FCM named in the DEV INTERLOCK SOURCE field determines whether the device is locked. When the result goes true the device is locked and the only possible command entry is the one specified in the PERMITTED COMMAND field.

Valid entries are:

YES	enable the interlock
NO	disable the interlock

**SPECIAL COMMAND Field**

This field specifies the command automatically sent to the device as part of transition override. See the description of the TRANSITION OVERRIDE field for an explanation of this feature.

Valid entry is:

a command from the descriptor set defined for the device on its Device Descriptor Templet

**SAFE COMMAND Field**

This field specifies the command automatically sent to the device as part of transition override or abnormal override. See the descriptions of the TRANSITION OVERRIDE field and the ABNORMAL OVERRIDE field for explanations of these features.

Valid entry is:

a command from the descriptor set defined for the device on its Device Descriptor Templet

**PERMITTED COMMAND Field**

This field specifies the only possible command sent to the device as part of device interlock. See the description of the DEV INTERLOCK ENABLE field for an explanation of this feature.

Valid entry is:

a command from the descriptor set defined for the device on its Device Descriptor Templet

**REVERT ENABLE Field**

This field specifies whether or not revert is enabled. When it is enabled, after the operator or a program changes the device command (DEV\_CMND) state, DEV\_CMND is automatically changed to the command specified in the REVERT COMMAND field. The revert command is an internal mechanism only and is not sent to the device, that is, no field changes.

Valid entries are:

YES	enable
NO	disable

**REVERT COMMAND Field**

This field specifies the command to which DEV\_CMND is automatically set as part of revert.

Valid entry is:

a command from the descriptor set defined for the device on its Device Descriptor Template

**CUTOUT ENABLE Field**

This field specifies whether cutout is enabled for the loop. Cutout allows an FCM of this or another loop to control the posting of the alarms.

Valid entry is:

YES	enable the cutout feature
NO	disable the cutout feature

**OUTPUT ERROR ACTION Field**

This field specifies the actions taken when the output driver fails.

Valid entries are:

NOTHING	continue normal processing
GOTO BACKUP	processing goes to the Backup Controller Module if one is present. This option shuts down CCF processing in a Multibus subsystem when the driver fails.
LOOP OFF	turn the loop off

**LOG STATE CHANGES Field**

This field provides the means to enable or disable logging of state changes for this device (that is, an audit trail of device state change messages in a historical archive).

Valid entries are:

YES	log state changes for this device. Each time the device loop is scanned the old and new states are determined. If the old and new
-----	---

states differ, an attempt is made to send a state change message to the Alarm/Event package which sends the message to the History node. If memory cannot be allocated for the state change message, the History node will display, NOCOMMAND in place of the missing device states. This indicates to the operator that at least one state change was not logged.

NO do not log state changes for this device

**DMND PROCESSING MODE Field,  
# LOOPS TO PRESCAN Field, and  
LOOPS TO PROCESS Field**

The DMND PROCESSING MODE field, # LOOPS TO PRESCAN field, and LOOPS TO PROCESS fields allow you to specify up to 5 loops to process along with this loop. The DMND PROCESSING MODE field is used to specify whether the loops are processed when this loop is normally processed or when it is demand scanned, or both. The # LOOPS TO PRESCAN field determines how many of the listed loops are scanned before this loop and how many are scanned after. The LOOPS TO PROCESS fields list the loops for this type of processing.

Valid entries for the DMND PROCESSING MODE field are:

NONE	no extra processing
NORMAL	process loops as specified in the LOOPS TO PROCESS fields when this loop is normally scanned
DEMAND	process loops as specified in the LOOPS TO PROCESS fields when this loop is demand scanned
BOTH	process loops as specified in the LOOPS TO PROCESS fields when this loop is demand or normally scanned



## PID Controller FCM

This section describes the active runtime fields on the controller FCM templets.

The PID controller FCM is treated in a different manner from other FCMs because the PID Controller FCM Templet can have children. These children are the Adaptive Gain and Adaptive Reset Templets. When you fill in a templet planning form for a PID controller, the entry you make to the ADAPTIVE CONTROL field determines whether you need to fill in the Adaptive Gain and/or Adaptive Reset templets.

If the entry to the ADAPTIVE GAIN field is:

- GAIN fill in an Adaptive Gain Templet
- RESET fill in an Adaptive Reset Templet
- BOTH fill in an Adaptive Gain and an Adaptive Reset Templet
- NONE no Adaptive Templets are needed

### **PROCESSING RATE Field**

The field indicates the multiple of the loop processing rate at which the controller algorithm is executed. If the default (1) is used, the FCM is processed each time the loop is processed. If the field is changed to 2, the FCM is processed every second time the loop is processed. Valid entry is an integer multiple of the loop processing rate specified on the Loop Definition Templet. For example, if the value of this field is 3 and the loop processing rate is 4 seconds, the controller algorithm is processed every 12 seconds.

### **BASE GAIN Field**

This field specifies the gain (proportional response) in a fixed gain controller. If no adaptive functions are turned on, the base gain is the active gain of the FCM, and the FCM operates as a fixed gain controller. When an adaptive gain function is turned on, all changes in active gain due to the algorithm for that function are based on the base gain. Valid entry is a decimal number.

**BASE RESET Field**

This field specifies the base value for the reset. If no adaptive functions are turned on, the base reset is the active reset of the FCM. When an adaptive function is turned on, all changes in active reset due to the algorithm for that function are based on the base reset. Valid entry is a decimal number.

**PREACT TIME Field**

This field specifies the Pre-act time value. Valid entry is a value set between 0 and 32 minutes.

**INTEGRAL TYPE Field**

This field specifies the type of integral (reset) control action. Valid entries are:

NONE        reset response is turned off.

STD         reset response is turned on and is operating with the standard algorithm. It is used with either adaptive gain or adaptive reset and/or external feedback. Manual reset must be turned off.

MICROSCAN - reset response is turned on and is operating with the Microscan algorithm. It is used with either adaptive gain or adaptive reset, but not with external feedback. Manual reset must be turned off.

**LIMITED OUTPUT MODES Field**

This field specifies whether the output limits are applied to the manual output, track output, and/or auto output signals. Valid entries are: NONE, MAN, AUTO, AUTO/MAN, TRK, MAN/TRK, AUTO/TRK, ALL

**CONTROLLER ACTION Field**

This field specifies whether the FCM output is direct or reverse acting. Valid entries are:

DIRECT - output of FCM increases as input to FCM increases

REVERSE - output of FCM decreases as input to FCM increases

**PROCESS HIGH VALUE Field**

This field gives the value of the process input corresponding to 100.0% of the range. Valid entry is a decimal value.

**PROCESS LOW VALUE Field**

This field gives the value of the process input corresponding to 0.0% of the range. Valid entry is a decimal value.

**INITIAL MODE Field**

This field specifies the output mode at start-up time. Valid entries are:

MAN - start in the Manual mode

AUTO - start in the Auto mode

**INITIAL OUTPUT Field**

This field specifies the value of the output at startup time. Valid entry is a value in percent.

**INITIAL SETPOINT Field**

This field indicates the value of the setpoint at start-up time. Valid entry is a value scaled in engineering units

**INITIAL SETPT MODE Field**

This field specifies the setpoint mode at start-up time. Valid entries are:

LOC - local setpoint mode

REM - remote setpoint mode

**LIMITED SETPT MODES Field**

This field specifies the setpoint modes to which the setpoint limits are applied. Valid entries are: NONE, LOC, TRK, REM, RMP, TRK/REM, TRK/RMP, REM/RMP,

LOC/TRK/REM, LOC/TRK/RMP, LOC/TRK, LOC/REM/RMP, LOC/REM, TRK/REM/RMP, LOC/RMP, ALL

#### **MANUAL RESET MODE Field**

This field specifies whether the manual reset control action is used and whether it is used for bumpless transfers. If it is specified as BALANCE, the value of the manual reset is automatically adjusted by the system to provide bumpless transfers when switching to the Auto output mode. Valid entries are:

OFF - manual reset response is turned off

BALANCE - manual reset response is enabled and used to provide bumpless transfers to Auto mode

NO BALANCE - manual reset response is enabled but not used to provide bumpless transfers

#### **MANUAL RESET VALUE Field**

This field specifies the initial value of the manual reset. Valid entry is a value between -127 and 127%.

#### **MAN RESET HIGH LIMIT Field**

This field specifies the high limit for the manual reset value. Valid entry is a value between -127 and 127%.

#### **MAN RESET LOW LIMIT Field**

This field specifies the low limit for the manual reset value. Valid entry is a value between -127 and 127%.

#### **TRACK ACTIVATE STATE Field**

This field specifies the state the track activator must be in for tracking to occur. Valid entries are: TRUE, FALSE.

**INTERACTIVE FORM Field**

This field specifies the method to multiply the proportional gain of the FCM by the integral. Valid entries are: YES (use interactive form), NO (use non-interactive form).

INTERACTIVE FORM: Output =  $(G + G * (\text{integral term})) * \text{derivative term}$

NONINTERACTIVE FORM: Output =  $(G + \text{integral term}) * \text{derivative term}$

**ERROR SQUARED OPTION Field**

This field specifies whether the integral calculation uses the error signal or square of the error signal. Valid entries are:

NO            integral calculation uses the error signal

YES           integral calculation uses the square of the error signal

**GAIN LIMIT Field**

This field defines a limit for the gain of the controller above which a low pass filter is activated. Valid entry is a value set between 0.0 and 125.0

**FF/FB MODE Field**

The feedforward and feedback options are specified in this field. Valid entries are:

FF            use feedforward

FB            use feedback

FF/FB        use feedforward and feedback

**EXT FEEDBACK ENABLE Field**

This field specifies whether the external feedback function is enabled. Valid entries are:

NO            feedback function is off

YES           feedback function is on

**BAD INPUTS ACCEPTED Field**

This field specifies whether an input with BAD data quality is treated as if the data quality is GOOD. Valid entries are:

- NO           do not accept bad inputs
- YES           accept bad inputs

**ACTION ON BAD INPUT Field**

This field specifies the action to take if the controller input signal goes to BAD data quality. Valid entries are:

- FCM OFF   controller FCM is turned off
- MAN OUT   controller FCM goes to the Manual output mode
- NONE       FCM continues processing according to its configuration

**ACTION ON BAD SETPT Field**

This field specifies the action to take if the setpoint signal goes to a BAD data quality. Valid entries are:

- NONE       no special action is taken
- MAN OUT   output mode is switched to Manual
- LOC SPT   setpoint mode goes to Local
- FCM OFF   FCM is turned off

**SETPOINT RETURN MODE Field**

This field specifies the setpoint mode to enter when the mode returns from Track. Valid entries are:

- PREVIOUS   go to the mode active before the setpoint went to Track
- LOCAL       go to the Local mode
- REMOTE     go to the Remote mode

**SETPT VALUE ON FAIL Field**

This field specifies the setpoint value in effect upon a remote link failure. Valid entry is a value in engineering units.

**CHANGE SETPT ON FAIL Field**

This field specifies whether a new setpoint value is supplied upon a remote link failure. Valid entries are:

NO	do not change setpoint
YES	change setpoint

**OUTPUT RETURN MODE Field**

This field specifies the output mode to enter when the mode returns from Track. Valid entries are:

PREVIOUS	go to the mode active before the output began tracking
MANUAL	go to the Manual mode
AUTO	go to the Auto mode

**OUTPUT MODE ON FAIL Field**

This field specifies the output mode to enter upon a remote link failure. Valid entries are:

PREVIOUS	go to the mode active before the remote link failure
MANUAL	go to the Manual mode
AUTO	go to the Auto mode
NO CHANGE	remain in the present mode

**OUTPUT VALUE ON FAIL Field**

This field specifies the output value effective upon a remote link failure if the entry in the CHANGE OUTPT ON FAIL field is YES. Valid entry is a value in percent.

**CHANGE OUTPUT ON FAIL Field**

This field specifies if a new output value is supplied upon a remote link failure. Valid entries are:

- |     |                                |
|-----|--------------------------------|
| NO  | do not change the output value |
| YES | change the output value        |

**LINK TIME-OUT Field**

This field specifies the initial time interval for the watchdog timer. The timer is used to determine whether a supervisory program can continue to control the loop. Valid entry is a number of seconds

**BALANCE MODE Field**

This field indicates whether the system can manipulate the remote setpoint ratio and/or the remote setpoint bias to prevent process bumps when the setpoint state is switched to remote. Valid entries are:

- |       |   |
|-------|---|
| RATIO | FCM automatically manipulates ratio value of remote setpoint to balance transitions to Remote       |
| BIAS  | controller automatically manipulates bias value of remote setpoint to balance transitions to Remote |
| BOTH  | the system can manipulate both bias and ratio   |
| OFF   | the balance feature is not used   |

**BIAS MODE Field**

This field specifies which remote setpoint bias value the FCM is presently using. The configured entry is the initial bias mode. During runtime, this field changes each time the mode is switched at the console. Valid entries are:

- |     |  |
|-----|--|
| REM | bias comes from the remote bias source                         |
| LOC | bias is from the local source (entered by operator at console) |
| OFF | controller does not use a bias signal                          |



**LOCAL BIAS Field**

This field specifies the local bias value for the FCM. The configured entry is the initial local bias. During runtime, you can enter the local bias value at the console. Valid entry is a decimal value.

**LIMITED BIAS MODES Field**

This field specifies the bias limits applied to the bias modes. Valid entries are:

NONE

LOC limits apply to the local bias mode

REM limits apply to the remote bias mode

LOC/REM limits apply to local and remote bias modes

**BIAS HIGH LIMIT Field**

This field specifies the high limit for the bias values. This limit only applies to the bias modes specified in the LIMITED BIAS MODES field. Valid entry is a percentage value.

**BIAS LOW LIMIT Field**

This field specifies the low limit for the bias values. This limit only applies to the bias modes specified in the LIMITED BIAS MODES field. Valid entry is a percentage value.

**RATIO MODE Field**

This field specifies which remote setpoint ratio value the controller is presently using. The configured entry is the initial ratio mode. During runtime, the contents of this field change each time the mode is switched from the console. Valid entries are:

REMOTE ratio comes from the remote ratio source

LOCAL ratio is from the local source (entered by operator at console)

OFF FCM does not use a ratio signal

**LOCAL RATIO Field**

This field specifies the local remote setpoint ratio value for the controller. The configured entry is the initial local ratio. During runtime, you can change the local ratio value at the console. Valid entry is a value in engineering units.

**LIMITED RATIO MODES Field**

This field specifies the ratio modes to which the ratio limits are applied. Valid entries are:

NONE

LOC            limits apply to the local ratio

REM            limits apply to the remote ratio

LOC/REM      limits apply to both modes

**RATIO HIGH LIMIT Field**

This field specifies the high limit for the ratio values. This limit only applies to the ratio modes specified in the LIMITED RATIO MODES field. Valid entry is a value in engineering units.

**RATIO LOW LIMIT Field**

This field specifies the low limit for the ratio values. This limit only applies to the ratio modes specified in the LIMITED RATIO MODES field. Valid entry is a value in engineering units.

**INC MINIMUM OUTPUT Field**

This field specifies the critical value for the incremental PID algorithm. The critical value is the smallest result of the algorithm that affects the output device. For instance, when pulse duration outputs drive the output device, the critical value is the smallest result of the PID algorithm that turns on a pulse duration output. When the result is less than the critical value, the value is stored and the result has no effect. On the next scan, the stored value is used to calculate a new result. Unused results are stored until the cumulative result is large enough to use.

The following equation is used to determine the critical value for a pulse duration output:

$$CV = \frac{(100 \times DO\_RATE)}{(H)}$$

where:

DO_RATE	is the resolution in milliseconds for the pulse digital outputs (as specified on the Controller or Multibus I/O Template)
H	is the 100% (HI_COUNT) value for the pulse digital outputs (as defined on the Pulse Duration Output Template)

When the critical value is zero, there is no adjustment for ineffective results. To make an adjustment for ineffective results, the integral (reset) control action must be turned on. Valid entry is a value, in engineering units, which must be greater than or equal to 0 (zero)

#### **INC LOWER DEADBAND Field**

This field specifies the lower boundary of the deadband for the incremental PID algorithm. Valid entry is a value, in engineering units, which must be less than or equal to 0 (zero)

#### **INC UPPER DEADBAND Field**

This field specifies the upper boundary of the deadband for the incremental PID algorithm. Valid entries are a value, in engineering units, which must be greater than or equal to 0 (zero)

#### **OUTPUT HIGH LIMIT Field**

This field specifies the high limit for the output signal. It is used to limit the output when the FCM is in the modes specified in the LIMITED OUTPUT MODES field. It is also the alarm limit if the entry in the OUTPUT HIGH ALARMS? field is YES. Valid entry is a number between -14.0 and 114.0%.

**OUTPUT LOW LIMIT Field**

This field specifies the low limit for the output signal. It is used to limit the output when the FCM is in the modes specified in the LIMITED OUTPUT MODES field. It is also the alarm limit if the entry in the OUTPUT LOW ALARMS? field is YES. Valid entry is a number between -14.0 and 114.0%.

**OUTPUT ALARM DB Field**

This field is used to define a deadband for the output alarm of this FCM. Valid entry is a floating point number in percent of range.

**OUTPUT RATE LIMIT Field**

This field specifies the highest allowed rate of change of the output signal per second. It is used when the FCM is in the modes specified in the LIMITED OUTPUT MODES field (except for MAN, OUTPUT RATE LIMIT is not applicable for manual mode). It is also the alarm limit if the entry in the OUTPUT RATE ALARMS? field is YES. Valid entry is a number between 0.0 and 128.0%.

**SETPOINT HIGH LIMIT Field**

This field specifies the high limit for the setpoint values. This limit only applies to the setpoint modes specified in the LIMITED SETPT MODES field. It is also used as an alarm limit when the entry in the SETPOINT HIGH ALARMS? field is YES. Valid entry is a decimal value in engineering units.

**SETPOINT LOW LIMIT Field**

This field specifies the low limit for the setpoint values. This limit only applies to the setpoint modes specified in the LIMITED SETPT MODES field. It is also used as an alarm limit when the entry in the SETPOINT LOW ALARMS? field is YES. Valid entry is a decimal value in engineering units.

**DEVIATION HIGH LIMIT Field**

This field specifies the high alarm limit for the deviation value (normalized measured value input to the PID FCM is greater than the PID setpoint by this

amount). It is used when the entry in the DEVIATION HI ALARMS? field is YES. Valid entry is a decimal value in engineering units.

**DEVIATION LOW LIMIT Field**

This field specifies the low alarm limit for the deviation value (normalized measured value input to the PID FCM is less than the PID setpoint by this amount). It is used when the entry in the DEVIATION LOW ALARMS? field is YES. Valid entry is a decimal value in engineering units.

**DEVIATION ALARM DB Field**

This field is used to define a deadband for the deviation alarm of this FCM. Valid entry is a value in engineering units.

**OUTPUT HIGH ALARMS Field**

This field specifies whether the high alarm on the output signal is enabled. If this field is YES, an alarm occurs when the output is equal to or greater than the value entered in the OUTPUT HIGH LIMIT field. The alarm clears when the output decreases to a value equal to or less than the value of the high limit minus the alarm deadband. Valid entries are: YES (enable the alarm), NO (disable the alarm).

**OUTPUT LOW ALARMS Field**

This field specifies whether the low alarm on the output signal is enabled. If this field is YES, an alarm occurs when the output is equal to or less than the value entered on the OUTPUT LOW LIMIT field. The alarm clears when the output increases to a value equal to or greater than the value of the low limit plus the alarm deadband. Valid entries are: YES (enable the alarm), NO (disable the alarm).

**OUTPUT RATE ALARMS Field**

This field specifies whether the rate alarm for the output signal is enabled. If this field is YES, an alarm occurs when the rate of change of the output per second is equal to or greater than the value entered in the OUTPUT RATE LIMIT field. Valid entries are: YES (enable the alarm), NO (disable the alarm).

**BAD OUTPUT ALARMS Field**

This field specifies whether the bad data quality alarm for the output signal is enabled. If this field is YES, an alarm occurs when the data quality of the output goes to BAD. Valid entries are: YES (enable the alarm), NO (disable the alarm).

**SETPOINT HIGH ALARMS Field**

This field specifies whether the high alarm for the setpoint signal is enabled. If this field is YES, an alarm occurs when the output is equal to or greater than the value entered in the SETPOINT HIGH LIMIT field. Valid entries are: YES (enable the alarm), NO (disable the alarm).

**SETPOINT LOW ALARMS Field**

This field specifies whether the low alarm for the setpoint signal is enabled. If this field is YES, an alarm occurs when the setpoint is equal to or less than the value entered in the SETPOINT LOW LIMIT field. Valid entries are: YES (enable the alarm), NO (disable the alarm).

**BAD SETPOINT ALARMS Field**

This field specifies whether the bad data quality alarm for the setpoint signal is enabled. If this field is YES, an alarm occurs when the data quality of the setpoint goes to BAD. Valid entries are: YES (enable the alarm), NO (disable the alarm).

**DEVIATION HI ALARMS Field**

This field specifies whether the high alarm for the deviation signal is enabled. If this field is YES, an alarm occurs when the deviation is equal to or greater than the value entered in the DEVIATION HIGH LIMIT field. Valid entries are: YES (enable the alarm), NO (disable the alarm).

**DEVIATION LOW ALARMS Field**

This field specifies whether the low alarm for the deviation signal is enabled. If this field is YES, an alarm occurs when the deviation is equal to or greater than the value entered on the DEVIATION LOW LIMIT field. Valid entries are: YES (enable the alarm), NO (disable the alarm).

**Alarm PRIORITY Fields**

These following fields specify the priority levels for their respective alarms.

- OUTPUT HIGH PRIORITY
- OUTPUT LOW PRIORITY
- OUTPUT RATE PRIORITY
- BAD OUTPUT PRIORITY
- SETPOINT HIGH PRIORITY
- SETPOINT LOW PRIORITY
- BAD SETPT PRIORITY
- DEVN HIGH PRIORITY
- DEVN LOW PRIORITY

Valid entries are:

STD	standard priority
MED	medium priority
HIGH	high priority

## Adaptive Gain

The PID Controller FCM uses adaptive gain and/or adaptive reset to adapt to changing process conditions. This templet is a child of the PID Controller FCM Templet.

**HIGH LIMIT Field**

The field specifies the overall high limit for gain or reset when any adaptive function is used. Valid entry is a decimal value.

**LOW LIMIT Field**

This field specifies the overall low limit for gain or reset when any adaptive function is used. Valid entry is a decimal value.

**CONTACT STATE Field**

This field specifies on which state of the contacts the gain or reset is adapted by the contact adaptive function.

Valid entries are:

- |       |  |
|-------|--|
| TRUE  | apply the contact adaptive function when the signal is true (closed contacts - non-zero) |
| FALSE | apply the contact adaptive function when the signal is false (open contacts - zero)      |

**CONTACT GAIN Field**

This field specifies the factor by which the base gain or reset is multiplied when the contact adaptive function is applied. Valid entry is a decimal value.

**REMOTE HIGH LIMIT Field**

This field specifies the maximum allowed value for the remote variable adaptive contribution to the overall gain or reset. Valid entry is a decimal number.

**REMOTE LOW LIMIT Field**

This field specifies the minimum allowed value for the remote variable adaptive contribution to the overall gain or reset. Valid entry is a decimal number.

**REMOTE UPPER BRKPT Field**

This field specifies the upper breakpoint, BP2, for the remote variable adaptive function. The gain or reset changes when the remote variable becomes greater than the upper breakpoint.

Valid entry is a decimal value with the same type of units as the remote variable (eng. units or percent).



**REMOTE LOWER BRKPT Field**

This field specifies the lower breakpoint, BP1, for the remote variable adaptive function. The gain or reset changes when the remote variable becomes less than the lower breakpoint.

Valid entry is decimal value with same type of units as the remote variable (engineering units or percent).

**REMOTE UPPER FACTOR Field**

This field specifies the multiplier of the base gain or reset associated with the upper breakpoint, BP2, of the remote variable adaptive function. Valid entry is a decimal value.

**REMOTE LOWER FACTOR Field**

This field specifies the multiplier of the base gain or reset associated with the lower breakpoint, BP1, of the remote variable adaptive function. Valid entry is a decimal value.

**OUTPUT UPPER BRKPT Field**

This field specifies the upper breakpoint, BP2, for the output adaptive function. The gain or reset starts to change when the size of the output becomes greater than the upper breakpoint. Valid entry is a decimal value in percent.

**OUTPUT LOWER BRKPT Field**

This field specifies the lower breakpoint, BP1, for the output adaptive function. The gain or reset starts to change when the size of the output becomes less than the lower breakpoint. Valid entry is a decimal value in percent.

**OUTPUT UPPER FACTOR Field**

This field specifies the multiplier of the base gain or reset associated with the upper breakpoint, BP2, of the output adaptive function. Valid entry is a decimal value.

**OUTPUT LOWER FACTOR Field**

This field specifies the multiplier of the base gain or reset associated with the lower breakpoint, BP1, of the output adaptive function. Valid entry is a decimal value.

**PROCESS HIGH LIMIT Field**

This field specifies the maximum allowed value for the process adaptive contribution to the overall gain or reset. Valid entry is a decimal number.

**PROCESS LOW LIMIT Field**

This field specifies the minimum allowed value for the process adaptive contribution to the overall gain or reset. Valid entry is low limit value.

**PROCESS UPPER BRKPT Field**

This field specifies the upper breakpoint, BP2, for the process adaptive function. The gain or reset starts to change when the size of the process variable becomes greater than the upper breakpoint. Valid entry is a decimal value in percent.

**PROCESS LOWER BRKPT Field**

This field specifies the lower breakpoint, BP1, for the process adaptive function. The gain or reset changes when the process variable becomes less than the lower breakpoint. Valid entry is a decimal value in percent.

**PROCESS UPPER FACTOR Field**

This field specifies the multiplier of the base gain or reset associated with the upper breakpoint, BP2, of the process adaptive function. Valid entry is a decimal value.

**PROCESS LOWER FACTOR Field**

This field specifies the multiplier of the base gain or reset associated with the lower breakpoint, BP1, of the process adaptive function. Valid entry is a decimal value.

**ERROR HIGH LIMIT Field**

This field specifies the maximum allowed value for the error adaptive contribution to the overall gain or reset. Valid entry is the high limit value.

**ERROR LOW LIMIT Field**

This field specifies the minimum allowed value for the error adaptive contribution to the overall gain or reset. Valid entry is the low limit value.

**ERROR UPPER BRKPT Field**

This field specifies the upper breakpoint, BP2, for the error adaptive function. The gain or reset changes when the error becomes greater than the upper breakpoint. Valid entry is a decimal value in percent.

**ERROR LOWER BRKPT Field**

This field specifies the lower breakpoint, BP1, for the error adaptive function. The gain or reset changes when the error becomes less than the lower breakpoint. Valid entry is a decimal value in percent.

**ERROR UPPER FACTOR Field**

This field specifies the multiplier of the base gain or reset associated with the upper breakpoint, BP2, of the error adaptive function. Valid entry is a decimal value.

**ERROR LOWER FACTOR Field**

This field specifies the multiplier of the base gain or reset associated with the lower breakpoint, BP1, of the error adaptive function. Valid entry is a decimal value.

## Adaptive Reset

The PID Controller FCM uses adaptive gain and/or adaptive reset to adapt to changing process conditions. This template is a child of the PID Controller FCM Template.

See field description under [Adaptive Gain](#) on page 247.

## Auto/Manual Controller FCM

The following tunable fields are the same as in the PID Controller templet:

PROCESSING RATE Field	ACTION ON BAD INPUT Field
TRACK ACTIVATE STATE Field	BAD INPUTS ACCEPTED Field
INITIAL MODE Field	INITIAL OUTPUT Field
OUTPUT RETURN MODE Field	LINK TIME-OUT Field
OUTPUT MODE ON FAIL Field	OUTPUT VALUE ON FAIL Field
CHANGE OUTPUT ON FAIL Field	BALANCE MODE Field
BIAS MODE Field	LOCAL BIAS Field
LIMITED BIAS MODES Field	BIAS HIGH LIMIT Field
BIAS LOW LIMIT Field	RATIO MODE Field
LOCAL RATIO Field	LIMITED RATIO MODES Field
RATIO HIGH LIMIT Field	RATIO LOW LIMIT Field
LIMITED OUTPUT MODES Field	OUTPUT HIGH LIMIT Field
OUTPUT LOW LIMIT Field	OUTPUT ALARM DB Field
OUTPUT RATE LIMIT Field	OUTPUT HIGH ALARMS Field
OUTPUT LOW ALARMS Field	OUTPUT RATE ALARMS Field
BAD OUTPUT ALARMS Field	OUTPUT HIGH PRIORITY*
OUTPUT LOW PRIORITY*	OUTPUT RATE PRIORITY*
BAD OUTPUT PRIORITY*	*see <a href="#">Alarm PRIORITY Fields</a>

## Input FCMs

This section describes the active runtime fields on the input FCM templates.

### Common Fields

#### **INITIAL OUTPUT MODE Field**

This field determines the output mode for the FCM at startup. Valid entries are:

AUTO	FCM determines output values
MAN	operator determines output values

#### **INITIAL OUTPUT Field**

This field determines the result (output value of the FCM) in effect upon startup of the FCM. Valid entry is a decimal number.

### Analog Input FCM

#### **COUNTS - LOWER BOUND Field**

This field specifies the lower limit for the range of counts. It corresponds to the engineering units lower bound. Valid entry is an integer between 0 and 32767.

#### **COUNTS - UPPER BOUND Field**

This field specifies the upper limit for the range of raw counts. It corresponds to the engineering units upper bound. Valid entry is an integer between 0 and 32767.

#### **ENGU - LOWER BOUND Field**

This field specifies the engineering units range lower limit. Valid entry is a decimal number.

**ENGU - UPPER BOUND Field**

This field specifies the engineering units range upper limit. Valid entry is a decimal number. See the *CCF User's Guide* if the loop receives input from a smart device.

**Digital Input FCM****INVERTED INPUTS Field**

This field is used to modify the input bit patterns received by the Digital Input FCM. Valid entry is a decimal number converted to binary bit positions, specifying which bits are inverted

**Pulsed Input FCM****ENGU - LOWER BOUND Field**

This field specifies the lower limit of the engineering units range. Valid entry is a decimal number.

**ENGU - UPPER BOUND Field**

This field specifies the upper limit of the engineering units range. Valid entry is a decimal number.

**Pulsed Input Time Derivative FCM****CONSTANT Field**

This field specifies the scaling constant. You can choose this constant to express the result in pulses per minute or hour, and so on. Valid entry is a decimal number.

**BAD INPUTS ACCEPTED? Field**

Specifies whether an input with a BAD data quality is treated as if the data quality were GOOD. Valid entries are: NO, YES.

## Output FCMs

This section describes the active runtime fields on the output FCM templates.

### Common Fields

#### **INITIAL OUTPUT MODE Field or INITIAL MODE Field**

These fields indicate the output mode for the loop at start-up and the TRIO warmstart action. Valid entries are:

AUTO	loop determines output values
MAN	operator determines output values

#### **INITIAL OUTPUT Field**

This field determines the initial result (output of the FCM) at start-up of the FCM. Valid entry is a decimal number.

#### **BAD INPUTS ACCEPTED? Field**

This field specifies the data qualities acceptable for the input signal to the FCM and the action to take if the data quality becomes unacceptable. Valid entries are:

YES	GOOD and BAD are acceptable data qualities
NO	GOOD is the only acceptable data quality for the input

#### **ACTION ON ERROR Field**

This field defines the action taken if the FCM is unable to output a valid signal to the field. Valid entries are:

NOTHING	continue processing the loop as configured
FCM OFF	turn the FCM off
SOURCE OFF	turn the source off
GOTO BACKUP	for switchover to Backup Controller in Controller Subsystems with Backup Controller option. For other

Controller and Multibus I/O Subsystems, this choice shuts down CCF processing when an error condition occurs.

LOOP OFF      turn the loop off

## Analog Output FCM

### **SIGNAL INVERSION Field**

This field indicates whether the output signal is inverted before it is sent to the I/O Module. For details see INPUT SOURCE field in Section 8.2, Parameters Common To All Output FCMS.

Valid entries are:

YES            invert the signal  
NO             do not invert the signal

### **ENGU OR PERCENT Field**

This field specifies whether the FCM input is converted from engineering units or percent. This determines which scaling equation is used to change input to counts. Valid entries are:

ENGU         convert input from engineering units  
PERCENT     convert input from percent

### **COUNTS - LOWER BOUND Field**

This field specifies the raw counts lower limit corresponding to the engineering units lower bound. Valid entry is an integer between 0 and 32767.

### **COUNTS - UPPER BOUND Field**

This field specifies the raw counts upper limit corresponding to the engineering units upper bound. Valid entry is an integer between 0 and 32767



**ENGU - LOWER BOUND Field**

This field specifies the lower limit of the engineering units range. Valid entry is a decimal number

**ENGU - UPPER BOUND Field**

This field specifies the upper limit of the engineering units range. Valid entry is a decimal number.

**UNDER RANGE COUNTS Field and OVER RANGE COUNTS Field**

These fields indicate the lowest and highest count values that can be output. Valid entries are integers between 0 and 4095.

**Digital Output FCM****INVERTED OUTPUTS Field**

This field makes modifications to the bit patterns output by the Digital Output FCM. Valid entry is a decimal number translated into binary bit positions, specifying which bits are inverted.

**Pulse Duration Output FCM****COUNTS - UPPER BOUND Field**

This field defines the length for the pulse duration (Pulse Duration Output) or the number of pulses (Pulse Train Output) to treat as 100% of range. Valid entry is an integer for the pulse duration in milliseconds or an integer for the number of pulses.

**Pulse Train Output FCM****COUNTS - UPPER BOUND Field**

This field defines the length for the pulse duration (Pulse Duration Output) or the number of pulses (Pulse Train Output) to treat as 100% of range. Valid entry is an integer for the pulse duration in milliseconds or an integer for the number of pulses.

## Common Calculator FCM Fields

The following fields are common to most all calculator FCMs.

### **INITIAL MODE Field**

This parameter specifies if the output mode of the FCM is set at MAN (manual) or AUTO.

### **INITIAL OUTPUT Field**

This parameter specifies the result of the FCM calculations and can be changed if the output mode is manual. There are no state values nor state mnemonics available for TCL for this parameter.

### **BAD INPUTS ACCEPTED? Field**

This field specifies the data qualities acceptable for the input signal to the FCM and the action to take if the data quality becomes unacceptable. Valid entries are:

- |     |  |
|-----|--|
| YES | GOOD and BAD are acceptable data qualities             |
| NO  | GOOD is the only acceptable data quality for the input |

## Math Related Calculator FCMs

This section describes the additional active runtime fields on the following math related calculator FCM templates. See [Common Calculator FCM Fields](#) on page 258 for common fields.

- Sum of 4 Inputs
- Average
- Polynomial
- Subtraction
- Division
- Natural Logarithm
- Exponentiation
- Multiplication
- Absolute Value

### Sum of 4 Inputs FCM

The Sum of 4 Inputs FCM finds the sum of up to 4 input signals, each with an optional scaling constant.

#### **Constant 1 through Constant 4 Fields**

These fields specify the values of the constants. Valid entry is any number

### Polynomial FCM

The Polynomial FCM allows for polynomial curve fitting linearization for up to a third order fit.

#### **Constant 1 through Constant 4 Fields**

These fields specify the values of the constants. Valid entry is any number

### Average FCM

The Average FCM finds the arithmetic average of up to 4 inputs. You can use constant factors to scale the inputs.

**Constant 1 through Constant 4 Fields**

These fields specify the values of the constants. Valid entry is any number

**MINIMUM # INPUTS Field**

This field specifies the minimum number of inputs that must have a data quality of GOOD for the algorithm to calculate. Valid entries are: 2, 3, 4 or ALL INPUTS.

**Subtraction FCM**

The Subtraction FCM subtracts one input from another. You can use the constant factors for scaling.

**Constant 1 and Constant 2 Fields**

These fields specify the values of the constants. Valid entry is any number.

**Division FCM**

The Division FCM calculates the quotient of two input signals. You can use the constant factors for scaling.

**Constant 1 and Constant 2 Fields**

These fields specify the values of the constants. Valid entry is any number.

**Natural Logarithm FCM**

The Natural Logarithm FCM calculates a scaled natural logarithm for an optionally scaled input signal. You can use the constant factors for scaling.

**Constant 1 and Constant 2 Fields**

These fields specify the values of the constants. Valid entry is any number.

**Exponentiation FCM**

The Exponentiation FCM raises one input signal to a power determined by another input signal. You can use the constant factors for scaling.

**Constant 1 and Constant 2 Fields**

These fields specify the values of the constants. Valid entry is any number.

**Multiplication FCM**

The Multiplication FCM calculates the product of two inputs. You can also use constants to scale the product.

**Constant 1 Field**

This field specifies the values of the constant. Valid entry is any number

**Absolute Value FCM**

The Absolute Value FCM calculates the absolute value of the input signal.

**Scale Compensation Calculator FCMs**

This section describes the active runtime fields on the following scale compensation calculator FCM templates. See [Common Calculator FCM Fields](#) on page 258 for common fields.

- Modified Square Root
- Linearization
- Normalize
- Inverse Normalize
- Scale Input
- Flow Calculation
- Temperature Compensation
- Ratio/Bias

**Modified Square Root FCM**

The Modified Square Root FCM calculates the square root of its input for input values greater than 2.47% of range. For input values less than 2.47%, including negative input values, the result is modified to be linear near zero and to handle negative numbers. The Modified Square Root FCM extracts the square root and scales the percent value into engineering units.

**HI ENG UNITS Field (High Engineering Units)**

This field specifies the high limit of the engineering units range. Valid entry is any number.

**LO ENG UNITS Field (Low Engineering Units)**

This field specifies the low limit of the engineering units range. Valid entry is any number.

**Linearization FCM**

This FCM uses the table look-up method to linearize an input. The linearization tables support thermocouples and RTDs (Resistance Temperature Devices) used by the system. As an option, the FCM can linearize according to a table defined by you via the Breakpoint Sets Templet. The result of this FCM is scaled in degrees C (Celsius), F (Fahrenheit), K (Kelvin), or R (Rankine).

**INVERSE AXIS? Field**

This field specifies either normal or inverted axis. Valid entries are:

- |     |  |
|-----|--|
| NO  | do not invert axis (input value is the x value in the linearization table, corresponding y value becomes the result) |
| YES | invert axis (input value is the y value in the linearization table, corresponding x value becomes the result)        |

## Normalize FCM

The Normalize FCM normalizes a signal from an engineering unit scale to a dimensionless zero-based scale.

### **LOW RANGE Field**

This field specifies the low range value for the input signal. Valid entry is any number.

### **HIGH RANGE Field**

This field specifies the top of range value for engineering units range of the result. Valid entry is any number.

### **SCALING Field**

This field specifies the scaling factor. Valid entry is any number.

## Inverse Normalize FCM

The Inverse Normalize FCM converts a value from a normalized, dimensionless, zero-based scale (percentage) to a value on an engineering units scale.

### **LOW RANGE Field**

This field specifies the low range value for the input signal. Valid entry is any number.

### **HIGH RANGE Field**

This field specifies the top of range value for engineering units range of the result. Valid entry is any number.

### **SCALING Field**

This field specifies the scaling factor. Valid entry is any number.

## Scale Input FCM

The Scale FCM converts a value from one engineering unit scale to its corresponding value in another engineering unit scale.

### **OLD LOW RANGE Field**

This field specifies low range value of the input. Valid entry is any number.

### **OLD HIGH RANGE Field**

This field specifies high range value of the input. Valid entry is any number.

### **NEW LOW RANGE Field**

This field specifies low range value of the result. Valid entry is any number.

### **NEW HIGH RANGE Field**

This field specifies high range value of the result. Valid entry is any number.

## Flow Calculation FCM

This FCM calculates mass and volume flow rates for liquids and gases. Its primary input is either a differential pressure measurement or a velocity measurement.

### **STANDARD TEMP Field**

This field specifies the standard temperature. Temperature must be absolute (degrees Rankine). Valid entry is any number.

### **FLOWING TEMP Field**

This field specifies the flow temperature. It is used when TEMP LOC/REM is LOCAL. The temperature must be absolute (deg. Rankine where deg. Rankine = deg. Fahrenheit + 459.69). Valid entry is any number greater than 0.



**FLOWING PRESSURE Field**

This field specifies the flow pressure. It is used when PRES LOC/REM is LOCAL. Valid entry is any number greater than 0. The entry must be in PSIA.

**FLOWING DENSITY Field**

This field specifies the flow density. It is used when DEN LOC/REM is LOCAL. Valid entry is any number greater than 0.

**DENSITY SC Field**

This field specifies the value for the density at standard conditions. Valid entry is any number.

**SPEC GRAV SC Field**

This field specifies the value for specific gravity at standard conditions. Valid entry is any number.

**COEF THERM EXP Field**

This field specifies the value for the coefficient of thermal expansion. Valid entry is any number.

**FLOW COEFFICIENT Field**

This field specifies the value for the flow coefficient. Valid entry is any number.

**Temperature Compensation FCM**

The Temperature Compensation FCM calculates temperature compensated volumetric and mass flow rates for incompressible fluids.

**DES TEMP CONSTANT Field**

This field specifies the design temperature constant. Valid entry is any number.

**FLOW CONSTANT Field**

This field specifies the value of the flow constant. Valid entry is any number.

**DEN VS TEMP CONSTANT Field**

This field specifies the value of the density versus temperature constant. Valid entry is any number.

**VOLUME OR MASS FLOW? Field**

This field specifies whether the FCM uses volumetric flow rate or mass flow rate algorithm. Valid entries are:

MASS	use the mass flow rate algorithm
VOL	use the volumetric flow rate algorithm

**Ratio/Bias FCM**

The Ratio/Bias FCM is used to apply bias and ratio to a signal. Bias and ratio sources can be local or remote.

**LOCAL BIAS Field**

This field specifies the local bias value. It is used when the bias mode is LOCAL. Valid entry is any number.

**LOCAL RATIO Field**

This field specifies the local ratio value. It is used when the ratio mode is LOCAL. Valid entry is any number.

**Time Related Calculator FCMs**

This section describes the active runtime fields on the following time related calculator FCM templates. See [Common Calculator FCM Fields](#) on page 258 for common fields.

- Time Derivative

- Time Integration
- First Order Filter
- Totalizer
- Lead/Lag Filter
- Dead Time
- Dead Time Compensation
- Timer
- Counter
- Delay Timer

### Time Derivative FCM

The Time Derivative FCM performs a first order time differentiation on its input.

#### **CONSTANT Field**

This field specifies the value of the scaling constant. Valid entry is any number.

### Time Integration FCM

The Time Integration FCM performs time integration on its input. The calculation is a trapezoidal approximation of the integral.

#### **SCALE FACTOR Field**

This field specifies the value of the scaling constant. Valid entry is any number.

### First Order Filter FCM

The First Order Filter FCM provides a first order lag function.

#### **LAG TIME Field**

This field specifies the value of the lag time. Valid entry is an amount of time in minutes.

## Totalizer FCM

The Totalizer FCM calculates a total by adding or subtracting its current input from the total of the previous input values. You can also compare the total to two user-defined values.

### **SCALE FACTOR Field**

This field specifies the value of the scaling constant. Valid entry is any number.

### **TRIP POINT 1 Field**

This field specifies the value of PREDETERMINED REGISTER NO. 1. Valid entry is any number greater than or equal to 0.

### **TRIP POINT 2 Field**

This field specifies the value of the PREDETERMINED REGISTER NO. 2. Valid entry is a number greater than or equal to the TRIP POINT 1 entry.

### **UP OR DOWN Field**

This field specifies if the totalizer adds or subtracts each new input from the current total. Valid entries are:

- UP            totalizer adds input
- DOWN        totalizer subtracts input

### **THRESHOLD VALUE Field**

This field specifies the threshold value. If the absolute value of the input is lower than this value, the input is discarded without being applied to the total. Valid entry is a number greater than or equal to 0.

## Lead/Lag Filter FCM

The Lead/Lag Filter FCM provides the capability for combined lead/lag filter conditioning.

**LAG TIME Field**

This field specifies the value of the lag time. Valid entry is an amount of time in minutes.

**PREACT TIME Field**

This field specifies the value of Pre-act time. Valid entry is an amount of time in minutes.

**PREACT GAIN Field**

This field specifies the value of Pre-act gain. Valid entry is any number.

**Dead Time FCM**

The Dead Time FCM implements a pure dead time.

**Dead Time Compensation FCM**

When the process contains a dead time which threatens control loop stability, the Dead Time Compensation FCM provides a compensated process signal to the loop controller.

**MAX DEAD TIME Field**

This field specifies the value of the deadtime time interval. Valid entry is a number in seconds (the data base holds this number internally in milliseconds).

**LAG TIME Field**

This field specifies the value of the lag time. Valid entry is a number in minutes.

**PREACT TIME Field**

This field specifies the value of Pre-act time. Valid entry is a number in minutes.

**PREACT GAIN Field**

This field specifies the value of Pre-act gain. Valid entry is any number.

## Timer FCM

The Timer FCM provides for triggering an action after a specified elapsed time or at a specified time of day. Both single shot and periodic timing are supported.

### **PERIODIC TIME Field**

This field specifies period length. It is used by each type of timer configured for this FCM. Valid entries are in the format:

hh:mm:ss    time in terms of hour, minute, and second (the data base holds the number internally in seconds)

## Counter FCM

This is a basic timer whose result is either in seconds or a specified time interval. The Counter FCM has two inputs. One resets the count, the other enables or disables counting. You can compare the count to two configured reference values, or you can initialize the count to a configured value.

### **TIME UNITS (SECONDS) Field**

This field provides a means to scale the result into the desired time units. The result is usually calculated in seconds. If this is appropriate for the application, leave this field at its default value, 1. If another time unit is required, enter the value in this field. The internal result value (in milliseconds) is converted to seconds and then divided by the entry to become the final result of the FCM. For example, to specify minutes, enter 60. To specify hours, enter 3600. Valid entry is a decimal number greater than 0.

### **LOWER TIME LIMIT Field**

This field specifies a reference value that the count is compared to. When the count is less than the value in this field, the data base flag TIM\_EXP1 is 0. When the count is greater than or equal to the value in this field, TIM\_EXP1 is set to 1. Valid entry is a decimal number greater than or equal to 0.

**HIGHER TIME LIMIT Field**

This field specifies a reference value that the count is compared to. When the count is less than the value in this field, the data base flag TIM\_EXP2 is 0. When the count is greater than or equal to the value in this field, TIM\_EXP2 is set to 1. When TIM\_EXP2 is 1, the count is frozen at its current value and does not increment on subsequent loop scans. Valid entry is a decimal number greater than or equal to 0.

**Delay Timer FCM**

This is a delay timer whose result is ON or OFF. There is one input to the Delay Timer FCM whose value is also ON or OFF. Two timers are present in the algorithm. One is related to the time that the input is in the ON state. The other is related to the time the input is in the OFF state.

**Logic Calculator FCMs**

This section describes the active runtime fields on the following logic calculator FCM templets. See [Common Calculator FCM Fields](#) on page 258 for common fields.

- Logical AND
- Logical OR
- Logical NOT
- Exclusive OR
- Set/Reset Flip-Flop
- Real Compare (Signal Comparison)

**Logical AND**

The Logical AND FCM performs the AND function on up to 4 inputs.

**LOGICAL/BITWISE Field**

Logical	If all the defined inputs are not 0, Result = 1. If any of the defined inputs is 0, Result = 0
BIT	Bits 1 of all inputs are ANDed together to get Bit 1 of the result. Bits 2 are ANDed together, and so on. The result is a decimal

number equivalent to the bit pattern resulting from the ANDing. Bits in the inputs are shifted to the right or left prior to ANDing.

The inputs for bitwise type of FCM are decimal numbers translated into bit patterns by the FCM. For example, an input of 9 means that bits 0 and 4 are 1 and the rest of the bits are 0.

### **SHIFT COUNT 1 through 4 Fields**

These fields specify how many places the input bits are shifted. SHIFT COUNT 1 is for input 1, and so on. These fields are used when LOGICAL/BITWISE is BITWISE. Valid entries are integer values between 0 and 16.

### **SHIFT DIRECTION 1 through 4 Fields**

These fields specify whether input bits are shifted left or right, when LOGICAL/BITWISE is BITWISE. Valid entries are:

- RIGHT      shift input to right
- LEFT        shift input to left

## **Logical OR**

The Logical OR FCM performs the OR function on up to 4 inputs.

### **LOGICAL/BITWISE Field**

- Logical      If all the defined inputs are 0, Result = 0. If any of the defined inputs is 1, Result = 1
- BIT          Bits 1 of all inputs are ORed together to get Bit 1 of the result. Bits 2 are ORed together, and so on. The result is a decimal number equivalent to the bit pattern resulting from the ORing. Bits in the inputs are shifted to the right or left prior to ORing.

The inputs for bitwise type of FCM are decimal numbers translated into bit patterns by the FCM. For example, an input of 9 means that bits 0 and 4 are 1 and the rest of the bits are 0.



**SHIFT COUNT 1 through 4 Fields**

These fields specify how many places the input bits are shifted. SHIFT COUNT 1 is for input 1, and so on. These fields are used when LOGICAL/BITWISE is BITWISE. Valid entries are integer values between 0 and 16.

**SHIFT DIRECTION 1 through 4 Fields**

These fields specify whether input bits are shifted left or right, when LOGICAL/BITWISE is BITWISE. Valid entries are:

RIGHT	shift input to right
LEFT	shift input to left

**Logical NOT**

The Logical NOT FCM performs the NOT function on its input.

**LOGICAL/BITWISE Field**

Logical	If inp = 0, Result = 1. If inp is not 0, Result = 0.
BIT	Individual bits of input are NOTed.

The inputs for bitwise type of FCM are decimal numbers translated into bit patterns by the FCM. For example, an input of 9 means that bits 0 and 4 are 1 and the rest of the bits are 0.

**Exclusive OR**

The Exclusive OR FCM performs the exclusive OR function on two inputs.

**LOGICAL/BITWISE Field**

Logical In logical mode, all non-zero inputs = 1. The result is determined by the following truth table:

Inp1	Inp2	Result
0	0	0

0	1	1
1	0	1
1	1	0

BIT Bits 1 of the inputs are exclusively ORed together to get Bit 1 of the result, and so on.

## Set/Reset Flip-Flop

The result of the Set/Reset Flip-Flop FCM is determined according to the following truth table:

Inp1	Inp2	Result
0	0	prev
0	1	0
1	0	1
1	1	0

where

prev is the value of the result from the previous sample period

any non-zero input is treated as a 1

As long as inp2 (Reset) is 0, the result goes to 1 as soon as inp1 (Set) goes to 1. The result remains at 1 regardless of the subsequent state of inp1. When inp2 goes to 1, the result is forced to 0, and remains there as long as inp2 is 1.

## Real Compare FCM

This FCM compares two analog signals and provides a logical output if a specified condition is met.

### COMPENSATION CONSTANT Field

This field specifies the comparison constant, when INPUT SOURCE 2 is not specified. Valid entry is any number.

**COMPARE OPERATOR Field**

This field specifies the operator for the comparison constant. Valid entries are:

EQ	equal to
NE	not equal to
GT	greater than
GE	greater than or equal to
LT	less than
LE	less than or equal to

**NORMAL STATE Field**

This field specifies the normal state of the FCM's result. Valid entries are: 0, 1

## Miscellaneous Calculator FCMs

This section describes the active runtime fields on the following miscellaneous calculator FCM templates. See [Common Calculator FCM Fields](#) on page 258 for common runtime fields.

- Selector
- Put Generic Value
- Get Generic Value
- Select Next
- Redundant Signal Selector
- Input Limiter
- User Calculations
- Data Entry
- String
- User Math Block
- MOD30\_MOD300\_MAP

### Selector FCM

The Selector FCM can compare up to four input signals and choose one of them for the output.

**SELECTOR MODE Field**

This field configures the selector for serial, parallel or delta selection. Valid entries are:

SERIAL    serial selection  
PARALLEL parallel selection  
DELTA     delta selection

**SELECT OP 1 through 3 Fields**

These fields specify the type of comparisons used for the selections that the FCM can make. SELECT OP 1 should always be defined. The other two SELECT OPs are defined as needed. Valid entries are:

LOW        select low value  
CENTER    select center value. For two inputs, the smaller value is chosen.  
            For four inputs, the second largest is chosen.  
HIGH       select high value

**MIN NUMBER OF INPUTS Field**

This field specifies the minimum number of inputs with a GOOD data quality that must be present if the FCM is to perform its algorithm. Valid entries are: 1, 2, 3, 4, or ALL INPUTS.

**Put Generic Value FCM**

The Put Generic Value FCM writes its result to a user-specified destination parameter in the Controller, Multibus I/O, Data Processor, or Interface Subsystem containing the FCM. The result may come from either the INPUT SOURCE field or the CONSTANT TO PUT field. There are two ways to specify location of the destination parameter: OUTPUT DESTINATION or RELATION name. In most cases OUTPUT DESTINATION is recommended.

**CONSTANT TO PUT Field**

This field specifies a constant to put to the destination parameter. This field is used if no input source is defined in the INPUT SOURCE field. Valid entry is any decimal number.

**Get Generic Value FCM**

The Get Generic Value FCM acquires the value of a user-specified source parameter in the Controller, Multibus I/O, Data Processor, or Interface Subsystem containing the FCM. You can also obtain the data quality of the source parameter. You can also configure the FCM to acquire the current state of an alarm for a measured variable, a setpoint, a deviation, an output, and so on.

**ATTRIBUTE NAME Field**

This field specifies the mnemonic of the source parameter. An entry must be made to this field. Valid entry is a parameter mnemonic.

**DATAQUAL ATTRIBUTE Field**

This field specifies the parameter from which to obtain the data quality of the parameter named in the ATTRIBUTE NAME field. When this parameter is obtained it is stored in the data quality of the result parameter (DATAQUAL). Valid entry is a parameter mnemonic.

**ALARM CONDITION Field**

This field is used to get the current status of an alarm for the parameter named in the ATTRIBUTE NAME field. This field specifies the type of alarm to check. It must be appropriate for the entry in the ATTRIBUTE NAME field. For example, the SETPT HI entry for this field should only be used when the entry in the ATTRIBUTE NAME field is a setpoint. You can check only one alarm for a variable. The alarm status becomes the result of the Get FCM. Valid entries are:

NONE	do not obtain an alarm state
DQ BAD	measured variable bad data quality alarm
HIHI	measured variable high high alarm

HI	measured variable high alarm
LO	measured variable low alarm
LOLO	measured variable low low alarm
IROC HI	measured variable input rate of change alarm
OUT DQ B	output bad data quality alarm
OUT HI	output high alarm
OUT LO	output low alarm
OUT IRC	output rate of change alarm
SETPT DQ	setpoint bad data quality alarm
SETPT HI	setpoint high alarm
SETPT LO	setpoint low alarm
DEV DQ BD	deviation bad data quality alarm
DEV HI	deviation high alarm
DEV LO	deviation low alarm

## Select Next

This FCM compares two inputs and determines which FCM in the loop is processed next.

### **COMPARISON CONSTANT Field**

This field specifies the value for the comparison constant. Input 1 is compared to the comparison constant if no entry is made to the INPUT SOURCE 2 field. Valid entry is any number.

### **COMPARE OPERATOR Field**

This field specifies the value for the comparison constant. Valid entries are:

EQ	equal to
NE	not equal to

GT	greater than
GE	greater than or equal to
LT	less than
LE	less than or equal to

## Redundant Signal Selector

This FCM provides a backup signal from another source in the event that the primary signal data quality goes BAD.

## Input Limiter FCM

The Input Limiter FCM defines analog input high and low limits. When the input exceeds the high or low limit, the input value is clamped at the specified limit and a flag is set. The input remains clamped as long as it exceeds the limit. The flag remains set until the input crosses a configured deadband (below the high limit or above the low limit). Even after the input returns within the boundaries of the high and low limits and is no longer clamped, the flag remains set until the input crosses the deadband. This prevents the flag from being continuously set and reset if the input oscillates around one of the limits.

### HI LIMIT Field

This field specifies the value of the high limit. Valid entry is any number greater than the value of the low limit

### LO LIMIT Field

This field specifies the value of the low limit. Valid entry is any number less than the value of the high limit.

### ENABLE HIGH LIMIT Field

This field specifies whether the high limit is enabled. Valid entries are:

NO	high limit is disabled
YES	high limit is enabled

**ENABLE LOW LIMIT Field**

This field specifies whether the low limit is enabled. Valid entries are:

NO	low limit is disabled
YES	low limit is enabled.

**User Calculations FCM**

The User Calculations (UCAL) feature allows you to add user-defined algorithms to CCF processing. This feature requires two main steps:

- A TCL program to execute the algorithm must be edited, compiled, and loaded.
- A CCF loop with a UCAL FCM must be configured. When processed, the UCAL FCM starts the TCL program and makes variables available to it. The program can also pass variables back to the UCAL FCM.

**Data Entry FCM**

The Data Entry FCM allows you to store 20 decimal values in the data base for use by other applications.

**VALUE 1 through VALUE 20 Fields**

These fields specify the decimal values that the FCM can hold. Valid entries are decimal numbers. See the *CCF User's Guide* for details.

**String FCM**

The String FCM is a user-defined string of up to 40 characters. The string is set during configuration (not tuneable from the templet)

The string can be accessed by TCL programs via tag-FCM access (that is, '\$tag'-'STR'.SVALUE). You can also use it on custom displays. The string can be accessed even if the loop with the FCM is turned off. The result of the String FCM is a numerical value not directly related to the string.



## User Math Block FCM

The User Math Block FCM is used to create user-defined calculations for a combination of input variables and constants. The FCM is part of a loop in the same manner as other FCMs. When the FCM is executed, its calculation is a real number that becomes the result of the FCM.

### **CONSTANTS K1 through K7 Fields**

These fields are used to specify up to 7 constants: K1 through K7. You can view and change K1 through K4 during runtime via the Loop/FCM Display. You can only view K5 through K7, you cannot change them. Valid entry is a real number.

### **INITIAL RESULT Field**

This field specifies an initial result for the FCM. It is available for the first scan of the FCM. You can view and change the value during runtime via the Loop/FCM Display. Valid entry is a real number.

### **HIGH LIMIT Field**

This field specifies a high limit for the result of the FCM if the entry in the ENABLE HIGH LIMIT field is YES. The limit applies to the final result only, not intermediate calculations. When the result is limited, the HI\_FLAG attribute is set to 1. TCL can set events to watch this flag or you can acquire its value using Get FCMs. Valid entry is a real number.

### **LOW LIMIT Field**

This field specifies a low limit for the result of the FCM if the entry in the ENABLE LOW LIMIT field is YES. The limit applies to the final result only, not intermediate calculations. When the result is limited, the LO\_FLAG attribute is set to 1. TCL can set events to watch this flag or you can acquire its value using Get FCMs. Valid entry is a real number.

### **ENABLE HIGH LIMIT Field**

This field enables and disables the high alarm limit checking. Valid entries are: YES (enabled), NO (disabled).

**ENABLE LOW LIMIT Field**

This field enables and disables the low alarm limit checking. Valid entries are: YES (enabled), NO (disabled).

**MOD30\_MOD300\_MAP**

This FCM is used when the MOD 300 control system communicates with a MOD 30 Instrument. This FCM is addressed in detail in instruction book C120, *Taylor™ MOD 30™ Interface User's Guide*.

**Extended Processing FCM**

The Extended Processing FCMs perform mathematical functions on data gathered over an extended period of time. These FCMs provide information for historical, long-range evaluation of a process, rather than for direct operational control.

This section describes the active runtime fields on the following extended processing FCM templates. See below for common runtime fields.

- Continuous Moving Average FCM
- Standard Deviation FCM
- Non-Rate Periodic Total FCM
- Periodic Average FCM
- Periodic Maximum FCM
- Periodic Minimum FCM
- Periodic Rate Total FCM

**Common Fields****INITIAL MODE Field**

This parameter specifies if the output mode of the FCM is set at MAN (manual) or AUTO.

**INITIAL OUTPUT Field**

This parameter specifies the result of the FCM calculations and can be changed if the output mode is manual. There are no state values nor state mnemonics available for TCL for this parameter.

**BAD INPUTS ACCEPTED? Field**

This field specifies the data qualities acceptable for the input signal to the FCM and the action to take if the data quality becomes unacceptable. Valid entries are:

- |     |  |
|-----|--|
| YES | GOOD and BAD are acceptable data qualities             |
| NO  | GOOD is the only acceptable data quality for the input |

**SAMPLE RATE Field**

This field specifies the interval at which the FCM samples the input value. Valid entry is in the form:

hh:mm:ss the sample interval in hours, minutes, seconds

The SAMPLE RATE, PERIOD UNIT, and PERIOD SIZE fields must be configured so the calculation is performed on no more than 127 samples.

**PERIOD UNIT Field**

This field specifies the time units for the period. It is used with the PERIOD SIZE field to specify the period for the FCM. Section 10.4.7, Discussion, describes how the period is used by periodic Extended Processing FCMs. Valid entries are:

MONTHLY, LAST DAY, WEEKLY, DAILY, HOURLY, MINUTES

**PERIOD SIZE Field**

This field specifies the size of the period. It is used with the PERIOD UNIT field to specify the period of the FCM. Section 10.4.7, Discussion, describes how the period is used by periodic Extended Processing FCMs. Valid entry is an integer from 1 to 99.

**NUM OF INVALID SAMPLES Field**

This field specifies the maximum number of samples with a BAD data quality that can be present and still have output data quality GOOD. This field is not used when the BAD INPUTS ACCEPTED? field is set to YES. Valid entry is an integer from 0 to 127.

**Continuous Moving Average FCM**

This module determines the moving average of an input over a specified number of samples. The moving average is calculated in the standard mathematical fashion, that is, when a new sample is added to the calculation, the oldest sample is discarded. The result is always available, even if too few samples have been taken to make up the full complement specified at configuration.

**MAX BAD SAMPLES Field**

This field specifies the maximum number of bad samples allowable and present to still perform the calculation. Valid entry is any integer from 0 to the value entered on the NUMBER OF SAMPLES field.

**Standard Deviation FCM**

The Standard Deviation FCM calculates the standard deviation of the input value over a specified time. The period is synchronized with the real time clock (minutes, hours, and so on).

**Non-Rate Periodic Total FCM**

The Non-Rate Periodic Total FCM determines the total value of the input over a specified time period. The time period is synchronized with the real time clock (minutes, hours, and so on).

**Periodic Average FCM**

The Periodic Average FCM determines the arithmetic average of an input signal over a selected time period. The time period is synchronized with the real time clock (minutes, hours, shifts, and so on).

## Periodic Maximum FCM

The Periodic Maximum FCM determines the maximum value of an input signal over a specified time period. The period is synchronized with the real time clock (minutes, hours, shifts, and so on).

## Periodic Minimum FCM

The Periodic Minimum FCM determines the minimum value of an input signal over a specified time period. The period is synchronized with the real time clock (minutes, hours, and so on).

## Periodic Rate Total FCM

The Periodic Rate Total FCM measures an integrated total for an input over a specified time period. The time period is synchronized with the real time clock (minutes, hours, and so on).

# S800 I/O Runtime Templates

The S800 I/O runtime templates include:

- Analog Input Modules: [AI810](#), [AI890](#) on page 286
- Differential Analog Input Module: [AI820](#) on page 288
- RTD Input Module: [AI830](#) on page 291
- Thermocouple Input Module: [AI835](#) on page 294
- Analog Output Module: [AO810](#), [AO890](#) on page 297
- Bipolar Analog Output Module: [AO820](#) on page 300
- Digital Input Modules, DI810 and DI890: [DI810](#) on page 303
- Digital Input Module for AC 400: [DI814](#) on page 306
- Digital Input Module: [DI820](#) on page 307
- Digital Input Module: [DI821](#) on page 309
- Digital Output Module: [DO810](#) on page 311
- Digital Output Module: [DO814](#) on page 315
- Digital Output Module: [DO820](#) on page 317

## AI810, AI890

S800 I/O Analog Input Module templates. The fields described for the AI810 template are also used for the AI890 except as noted in the field entry descriptions.

### **ACTIVE 01 through 08**

If the channel is to be active enter a **YES**.

If the channel will not be used enter a **NO**.

The default value is YES.

### **AUTO DOWNLOAD**

To automatically download the configuration to the I/O module enter a **YES**. In the future, if the configuration is to be loaded locally enter a **NO**.

Default value is YES.

### **CLUSTER**

Enter the I/O device cluster of the station where this module is located. The cluster connected directly to the FCI is number zero (0). The number in this field is 0 -7.

Default value is 0.

### **DEAD BAND 01 through 08**

Enter the difference in successive signal count values before a new value will be sent.

The valid values are:

NONE, 0, 1, 3, 7, 15, 31, 63, 127, 255, 511, 1023, 2047, 4095, 8191 and 16383.

The default value is NONE.

### **FILTER TIME 01 through 08**

Enter the lowpass filter time constant (msec).

The possible range is:

20 to 65535.

The default value is 20 (20msec).

### **LABEL**

Enter a characteristic name to identify the AI810 or AI890 Module of the station. The name may contain up to 20 characters.



For the AI890 module, enter a description that includes the module type for future reference to help alleviate confusion with an AI890 being used in the AI810 templet.

### **LINEARIZATION 01 through 08**

If the channel needs to apply the square-root-law linearization, enter a **YES**; if not enter a **NO**.

The default is NO.

### **POSITION**

Enter a number from 1 to 12 representing the position of the I/O module relative to the FCI. The first I/O module directly next to the FCI is number 1.

Default value is 1.



If positions are to be used for spare MTUs, the spares must be in place so the rest of the I/O modules are numbered correctly. The spare MTU will use a position number when it is on the ModuleBus.

### **POWER LINE FREQUENCY**

Enter the power line frequency to filter out (notch) the signal.

The valid values are:  
16, 50, and 60.

The default value is 60.

**RANGE 01 through 08**

The valid values are:

0..10V (**not valid for AI890**)

0..20mA

The default range is 0..20mA.

**REPORT FAULTS 01 through 08**

If channel faults are to be reported, enter a **YES**; if not enter a **NO**.

The default value is YES.

**RESERVED OPTIONS**

Used to select I/O module type:

Enter:

0 (zero) for the AI810 Module

274333696 for the AI890 Module

The default value is 0 (zero).

**VALUE CYCLE TIME**

Enter the channel value update period (msec) over the fieldbus. Refer to the *S800 I/O User's Guide* for details of AF 100 loading in relation to cycle time.

The valid values are:

1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048 and 4096.

The default value is 128.

**AI820**

S800 I/O AI820 Differential Analog Input Module Template.

**ACTIVE 01 through 04**

If the channel is to be active enter a **YES**.

If the channel will not be used enter a **NO**.



The default value is YES.

### **AUTO DOWNLOAD**

To automatically download the configuration to the I/O module enter a **YES**. In the future, if the configuration is to be loaded locally enter a **NO**.

Default value is YES.

### **CLUSTER**

Enter the I/O device cluster of the station where this module is located. The cluster connected directly to the FCI is number zero (0). The number in this field is 0 -7.

Default value is 0.

### **DEAD BAND 01 through 04**

Enter the difference in successive signal count values before a new value will be sent.

The valid values are:

NONE, 0, 1, 3, 7, 15, 31, 63, 127, 255, 511, 1023, 2047, 4095, 8191 and 16383.

The default value is NONE.

### **FILTER TIME 01 through 04**

Enter the lowpass filter time constant (msec) for each channel.

The possible range is:

20 to 65535.

The default value is 20 (20msec).

### **LABEL**

Enter a characteristic name to identify the AI820 Module of the station. The name may contain up to 20 characters.

**LINEARIZATION 01 through 04**

If a channel needs to apply the square-root-law linearization, enter a **YES**; if not enter a **NO**.

The default is NO.

**POSITION**

Enter a number from 1 to 12 representing the position of the I/O module relative to the FCI. The I/O module directly next to the FCI is number 1.

Default value is 1.



If positions are to be used for spare MTUs, the spares must be in place so the rest of the I/O modules are numbered correctly. The spare MTU will use a position number when it is on the ModuleBus.

**POWER LINE FREQUENCY**

Enter the power line frequency to filter out (notch) the signal.

The valid values are:

16, 50, and 60.

The default value is 60.

**RANGE 01 through 04**

The valid values are:

-10..10V

-20..20mA

-5..5V

The default range is -20..20mA.

**REPORT FAULTS 01 through 04**

If channel faults are to be reported, enter a **YES**; if not enter a **NO**.

The default value is YES.

**RESERVED OPTIONS**

This field reserved for future use. A zero (0) should be entered as a default value.

**VALUE CYCLE TIME**

Enter the channel value update period (msec) over the fieldbus. Refer to the *S800 I/O User's Guide* for details of AF 100 loading in relation to cycle time.

The valid values are:

1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048 and 4096.

The default value is 128.

**AI830**

S800 I/O AI830 RTD Input Module Templet.

**ACTIVE 01 through 08**

If the channel is to be active enter a **YES**. If the channel will not be used enter a **NO**.

The default value is YES.

**AUTO DOWNLOAD**

To automatically download the configuration to the I/O module enter a **YES**. In the future, if the configuration is to be loaded locally enter a **NO**.

Default value is YES.

**CLUSTER**

Enter the I/O device cluster of the station where this module is located. The cluster connected directly to the FCI is number zero (0). The number in this field is 0 -7.

Default value is 0.

**DEAD BAND 01 through 08**

Enter the difference in successive signal count values before a new value will be sent.

The valid values are:

NONE, 0, 1, 3, 7, 15, 31, 63, 127, 255, 511, 1023, 2047, 4095, 8191 and 16383.

The default value is NONE.

### **FILTER 01 through 08**

Enter the lowpass filter time constant (in 10msec units) for each channel. Filter times shorter than the update time are not usable. Update time can be calculated as:

$$T_{\text{Update}} = 160\text{ms} + (n * 80\text{ms}) \quad n = \text{number of active channels } 1\dots 8$$

Therefore the  $T_{\text{Update}}$  Range = 240ms to 800ms

The default value is 10 (100msec).

The possible range is:

10 to 65535.

### **LABEL**

Enter a characteristic name to identify the AI830 Module of the station. The name may contain up to 20 characters.

### **POSITION**

Enter a number from 1 to 12 representing the position of the I/O module relative to the FCI. The I/O module directly next to the FCI is number 1.

Default value is 1.



If positions are to be used for spare MTUs, the spares must be in place so the rest of the I/O modules are numbered correctly. The spare MTU will use a position number when it is on the ModuleBus.

### **POWER LINE FREQUENCY**

Enter the power line frequency to filter out (notch) the signal.

The valid values are:

16, 50, and 60.

The default value is 60.

**RANGE 01 through 08**

The possible ranges are:

-80..80 C, -112..176 F  
-200..250 C, -328..482 F  
-200..850 C, -328..1562 F  
-60..180 C, -76..356 F  
-80..260 C, -112..500 F  
-100..260 C, -148..500 F  
0..400 OHMS.

The default range is -328..1562 F.

When the user saves the template, the sensor type is used to determine if the user entered a correct range option. If the user did not enter a correct range option, an error message is generated and a default correct range option provided. The following are the correct range options and defaults for each sensor type:

Pt100 => -80..80 C, -112..176 F, -200..250 C, -328..482 F, -200..850 C, -328..1562 F (corrected default)

Ni100 => -60..180 C, -76..356 F (corrected default)

Ni120 => -80..260 C, -112..500 F (corrected default)

Cu10 => -100..260 C, -148..500 F (corrected default)

Resistor => 0..400 OHMS (corrected default).

**REPORT FAULTS 01 through 08**

If channel faults are to be reported, enter a **YES**; if not enter a **NO**.

The default value is YES.

**RESERVED OPTIONS**

This field reserved for future use. A zero (0) should be entered as a default value.

**SENSOR TYPE 01 through 08**

Enter the type of sensor that will be connected to each channel.

The valid values are:  
PT100, NI100, NI120, CU10 and RESISTOR

The default entry is PT100.

### **VALUE CYCLE TIME**

Enter the channel value update period (msec) over the fieldbus. Refer to the *S800 I/O User's Guide* for details of AF 100 loading in relation to cycle time.

The valid values are:  
1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048 and 4096.

The default value is 128.

## **AI835**

S800 I/O AI835 Thermocouple Input Module Template.

### **ACTIVE 01 through 08**

If the channel is to be active enter a **YES**.  
If the channel will not be used enter a **NO**.

The default value is YES.

### **AUTO DOWNLOAD**

To automatically download the configuration to the I/O module enter a **YES**. In the future, if the configuration is to be loaded locally enter a **NO**.

Default value is YES.

### **CLUSTER**

Enter the I/O device cluster of the station where this module is located. The cluster connected directly to the FCI is number zero (0). The number in this field is 0 -7.

Default value is 0.

**DEAD BAND 01 through 08**

Enter the difference in successive signal count values before a new value will be sent.

The valid values are:

NONE, 0, 1, 3, 7, 15, 31, 63, 127, 255, 511, 1023, 2047, 4095, 8191 and 16383.

The default value is NONE.

**FILTER 01 through 08**

Enter the lowpass filter time constant (in 10msec units) for each channel. Filter times shorter than the update time are not usable. Update time can be calculated as:

$$T_{\text{Update}} = 160\text{ms} + (n * 80\text{ms}) \quad n = \text{number of active channels } 1 \dots 8$$

Therefore the  $T_{\text{Update}}$  Range = 240ms to 800ms

The possible range is:

10 to 65535.

The default value is 10 (100msec).

**JUNCTION COMP 01 through 08**

Enter if the junction compensation will be from an entered fixed value or from the Cold Junction Compensation channel. If channel 08 will be used as the Cold Junction Compensation value, enter **COLD** for channels 01 through 07, and for channel 08 also enter **COLD**. If the junction compensation will use the value entered in the JUNCTION TEMP field, enter **FIXED**.

The valid values are:

COLD or FIXED

The default entry is COLD.

**JUNCTION TEMPERATURE**

Enter the fixed temperature at the junction of the thermocouple wire and the terminals of the MTU.

Enter the estimated temperature at the MTU terminals if Cold Junction Compensation will not be used by any one or all channels. If Cold Junction

Compensation will be used by all channels (01 to 07), then this field entry does not need a value, leave it at the default of 0.

The possible entries are:

If the entry of the TEMP UNIT = C: -40...100

If the entry of the TEMP UNIT = F: -40...212

The default entry is 0 (zero)

### **LABEL**

Enter a characteristic name to identify the AI835 Module of the station. The name may contain up to 20 characters.

### **POSITION**

Enter a number from 1 to 12 representing the position of the I/O module relative to the FCI. The I/O module directly next to the FCI is number 1.

Default value is 1.



If positions are to be used for spare MTUs, the spares must be in place so the rest of the I/O modules are numbered correctly. The spare MTU will use a position number when it is on the ModuleBus.

### **POWER LINE FREQUENCY**

Enter the power line frequency to filter out (notch) the signal.

The valid values are:

16, 50, and 60.

The default value is 60.

### **REPORT FAULTS 01 through 08**

If channel faults are to be reported, enter a **YES**; if not enter a **NO**.

The default value is YES.



**RESERVED OPTIONS**

This field reserved for future use. A zero (0) should be entered as a default value.

**TC TYPE 01 through 08**

Enter the type of sensor that will be connected to each channel. When channel 08 serves as the Cold Junction Compensation value, enter the sensor type, normally **PT100**, that will be connected to channel 8. The type **LINEAR** is used if the input is a -25mV to 75mV signal.

The valid values are:

B, C, E, J, K, N, R, S, T, PT100 and LINEAR

The default entry is J.

**TEMPERATURE UNIT**

Enter the temperature's unit of measure; either Centigrade or Fahrenheit.

The valid values are:

C and F

The default value is C.

**VALUE CYCLE TIME**

Enter the channel value update period (msec) over the fieldbus. Refer to the *S800 I/O User's Guide* for details of AF 100 loading in relation to cycle time.

The valid values are:

1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048 and 4096.

The default value is 128.

**AO810, AO890**

The S800 I/O AO810 Analog Output Module templet is also used for the AO890 except as noted in the field entry descriptions.

**ACTIVE 01 through 08**

If the channel is to be active enter a **YES**. If the channel will not be used enter a **NO**.  
The default value is YES.

**AUTO DOWNLOAD**

To automatically download the configuration to the I/O module enter a **YES**. In the future, if the configuration is to be loaded locally enter a **NO**.  
Default value is YES.

**CLUSTER**

Enter the I/O device cluster of the station where this module is located. The cluster connected directly to the FCI is number zero (0). The number in this field is 0 -7.  
Default value is 0.

**DQ CYCLE TIME**

Enter the channel data quality update period (msec) over the fieldbus. Refer to the *S800 I/O User's Guide* for details of AF 100 loading in relation to cycle time.

The valid values are:

1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, and 4096

The default value is 128.

**LABEL**

Enter a characteristic name to identify the AO810 or AO890 Module of the station.  
The name may contain up to 20 characters.



For the AO890 module, enter a description that includes the module type for future reference to help alleviate confusion with an AO890 being used in the AO810 templet.

**OSP HOLD 01 through 08**

If an OSP TIMEOUT has occurred, will the AO810 hold the last value during OSP? Enter a **YES** if it will, or enter a **NO** if an OSP VALUE will be used.

The Default is NO.

**OSP TIMEOUT**

Enter the time (msec) after communications is lost, that outputs go to OSP value.

The valid values are:

OFF, 256, 512, and 1024.

The default value is 1024.

**OSP VALUE 01 through 08**

Enter the channel value in counts to output after an OSP TIMEOUT. If the OSP HOLD has been selected as the OSP value then this field entry is ignored.

The valid values are:

0 to 28480.

Where 0 = 0 mA and 28480 = 20 mA (Counts per mA = 1424)

The default is 0.

**POSITION**

Enter a number from 1 to 12 representing the position of the I/O module relative to the FCI. The I/O module directly next to the FCI is number 1.

Default value is 1.



If positions are to be used for spare MTUs, the spares must be in place so the rest of the I/O modules are numbered correctly. The spare MTU will use a position number when it is on the ModuleBus.

**REPORT FAULTS 01 through 08**

If channel faults are to be reported, enter a **YES**; if not enter a **NO**.

The default value is YES.

**RESERVED OPTIONS**

Used to select I/O module type:

Enter:

0 (zero) for the AO810 Module

274333696 for the AO890 Module

The default value is 0 (zero).

**VALUE CYCLE TIME**

Enter the channel value update period (msec) over the fieldbus. Refer to the *S800 I/O User's Guide* for details of AF 100 loading in relation to cycle time.

The valid values are:

1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048 and 4096.

The default value is 128.

**AO820**

The S800 I/O AO820 Bipolar Analog Output Module template.

**ACTIVE 01 through 04**

If the channel is to be active enter a **YES**.

If the channel will not be used enter a **NO**.

The default value is YES.

**AUTO DOWNLOAD**

To automatically download the configuration to the I/O module enter a **YES**. In the future, if the configuration is to be loaded locally enter a **NO**.

Default value is YES.

**CLUSTER**

Enter the I/O device cluster of the station where this module is located. The cluster connected directly to the FCI is number zero (0). The number in this field is 0 -7.

Default value is 0.

### **DQ CYCLE TIME**

Enter the channel data quality update period (msec) over the fieldbus. Refer to *S800 I/O User's Guide* for details of AF 100 loading in relation to cycle time.

The valid values are:

1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, and 4096

The default value is 128.

### **LABEL**

Enter a characteristic name to identify the AO820 Module of the station. The name may contain up to 20 characters.

### **OSP HOLD 01 through 04**

If an OSP TIMEOUT has occurred, will the AO810 hold the last value during OSP? Enter a **YES** if it will, or enter a **NO** if an OSP VALUE will be used.

The Default is NO.

### **OSP TIMEOUT**

Enter the time (msec) after communications is lost, that outputs go to OSP value.

The valid values are:

OFF, 256, 512, and 1024.

The default value is 1024.

### **OSP VALUE 01 through 04**

Enter the channel value in counts to output after an OSP TIMEOUT. If the OSP HOLD has been selected as the OSP value then this field entry is ignored.

The default is 0.

The valid values are:

0 to 28480.

Where 0 = 0 mA and 28480 = 20 mA.

Output Range	Minimum Count	Maximum Count	Counts
-10 - 10V	-28480	28480	2848 V/count
-20 - 20MA	-28480	28480	1424 mA/count

### POSITION

Enter a number from 1 to 12 representing the position of the I/O module relative to the FCI. The I/O module directly next to the FCI is number 1.

Default value is 1.



If positions are to be used for spare MTUs, the spares must be in place so the rest of the I/O modules are numbered correctly. The spare MTU will use a position number when it is on the ModuleBus.

### RANGE 01 through 04

Enter the output range (voltage or current) desired.

The valid values are:

-10..10V

-20..20MA

The default value is -20..20MA

### REPORT FAULTS 01 through 04

If channel faults are to be reported, enter a **YES**; if not enter a **NO**.

The default value is YES.

### RESERVED OPTIONS

This field reserved for future use. A zero (0) should be entered as a default value.

**VALUE CYCLE TIME**

Enter the channel value update period (msec) over the fieldbus. Refer to the *S800 I/O User's Guide* for details of AF 100 loading in relation to cycle time.

The valid values are:

1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048 and 4096.

The default value is 128.

**DI810**

The DI810 templet is also used for the DI890 except as noted in the field entry descriptions. The DI890 also supports only 8 channels.

**ACTIVE 01 through 16**

If the channel is to be active enter a **YES**.

If the channel will not be used enter a **NO**.

The default value is YES.

**AUTO DOWNLOAD**

To automatically download the configuration to the I/O module enter a **YES**. In the future, if the configuration is to be loaded locally enter a **NO**.

Default value is YES.

**CLUSTER**

Enter the I/O device cluster of the station where this module is located. The cluster connected directly to the FCI is number zero (0). The number in this field is 0 -7.

Default value is 0.

**FILTER**

Enter the input filter time in msec.

The valid values are:

2, 4, 8 and 16.

The default value is 2.

### **LABEL**

Enter a characteristic name to identify the DI810 or DI890 Module of the station. The name may contain up to 20 characters.



For the DI890 module, enter a description that includes the module type for future reference to help alleviate confusion with an DI890 being used in the DI810 templet.

### **MONITOR POWER**

To monitor process power source enter a **YES**; if not needed enter a **NO**.

DI890 with any channel supervision requires that **MONITOR POWER = YES**.

Default Value is **NO**.

### **POSITION**

Enter a number from 1 to 12 representing the position of the I/O module relative to the FCI. The I/O module directly next to the FCI is number 1.

Default value is 1.

### **REPORT FAULTS 01 through 16**

If channel faults are to be reported, enter a **YES**;  
if not enter a **NO**.

The default value is **YES**.

### **RESERVED OPTIONS**

Used to select I/O module type:

Enter:

0 (zero) for the DI810 Module.

542769152 for the DI890 Module (decimal value for all channels without supervision) see table below where a "1" sets supervision for channels 8 - 1.



The default value is 0 (zero).

HEX Bytes	Byte 3 Option Code	Byte 2 Module Type No.	Byte 1 Channels 16 - 9 (bits)								Byte 0 Channels 8 - 1 (bits)								
Supervised Channels	b7 - b0	b7 - b0	b7	b6	b5	b4	b3	b2	b1	b0	b7	b6	b5	b4	b3	b2	b1	b0	
None	\$20	\$5A	Channels not used - set = 00 (zeros)								0	0	0	0	0	0	0	0	0
All	\$20	\$5A									1	1	1	1	1	1	1	1	
4 - 1	\$20	\$5A									0	0	0	0	1	1	1	1	
8 - 5	\$20	\$5A									1	1	1	1	0	0	0	0	
7, 2, 1	\$20	\$5A									0	1	0	0	0	0	1	1	
etc.	\$20	\$5A																	

Examples from above:

#### All with No Supervision

hex number = **205A0000**

converted to decimal = **542769152**

#### All with Supervision

hex number = **205A00FF**

converted to decimal = **542769407**

#### Channels 4 through 1 with Supervision

hex number = **205A000F**

converted to decimal = **542769167**

#### Channels 8 through 5 with Supervision

hex number = **205A00F0**

converted to decimal = **542769392**

#### Channels 7, 2 and 1 with Supervision

hex number = **205A0043**

converted to decimal = **542769219**

### VALUE CYCLE TIME

Enter the channel value update period (msec) over the fieldbus. Refer to *S800 I/O User's Guide* for details of AF 100 loading in relation to cycle time.

The valid values are:

1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048 and 4096.

The default value is 128.

## **DI814**

The DI814 templet on an AC 400 Series controller.

### **ACTIVE 01 through 16**

If the channel is to be active enter a **YES**.

If the channel will not be used enter a **NO**.

The default value is YES.

### **AUTO DOWNLOAD**

To automatically download the configuration to the I/O module enter a YES. In the future, if the configuration is to be loaded locally enter a NO.

Default value is YES.

### **CLUSTER**

Enter the I/O device cluster of the station where this module is located. The cluster connected directly to the FCI is number zero (0). The number in this field is 0 -7.

Default value is 0.

### **FILTER**

Enter the input filter time in msec.

The valid values are:

2, 4, 8 and 16.

The default value is 2.

### **LABEL**

Enter a characteristic name to identify the DI814 Module of the station. The name may contain up to 20 characters.

**MONITOR POWER**

To monitor process power source enter a **YES**;  
if not needed enter a **NO**.

Default Value is NO.

**POSITION**

Enter a number from 1 to 12 representing the position of the I/O module relative to the FCI. The I/O module directly next to the FCI is number 1.

Default value is 1.

**REPORT FAULTS 01 through 16**

If channel faults are to be reported, enter a **YES**;  
if not enter a **NO**.

The default value is YES.

**RESERVED OPTIONS**

This field reserved for future use. A zero (0) should be entered as a default value.

**VALUE CYCLE TIME**

Enter the channel value update period (msec) over the fieldbus. Refer to *S800 I/O User's Guide* for details of AF 100 loading in relation to cycle time.

The valid values are:

1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048 and 4096.

The default value is 128.

**DI820****ACTIVE 01 through 08**

If the channel is to be active enter a **YES**.  
If the channel will not be used enter a **NO**.

The default value is YES.

### **AUTO DOWNLOAD**

To automatically download the configuration to the I/O module enter a YES. In the future, if the configuration is to be loaded locally enter a NO.

Default value is YES.

### **CLUSTER**

Enter the I/O device cluster of the station where this module is located. The cluster connected directly to the FCI is number zero (0). The number in this field is 0 -7.

Default value is 0.

### **FILTER**

Enter the input filter time in msec.

The valid values are:

2, 4, 8 and 16.

The default value is 2.

### **LABEL**

Enter a characteristic name to identify the DI820 Module of the station. The name may contain up to 20 characters.

### **MONITOR POWER**

To monitor process power source enter a **YES**;  
if not needed enter a **NO**.

Default Value is NO.

### **POSITION**

Enter a number from 1 to 12 representing the position of the I/O module relative to the FCI. The I/O module directly next to the FCI is number 1.

Default value is 1.

### **REPORT FAULTS 01 through 08**

If channel faults are to be reported, enter a **YES**;  
if not enter a **NO**.

The default value is YES.

### **RESERVED OPTIONS**

This field reserved for future use. A zero (0) should be entered as a default value.

### **VALUE CYCLE TIME**

Enter the channel value update period (msec) over the fieldbus. Refer to *S800 I/O User's Guide* for details of AF 100 loading in relation to cycle time.

The valid values are:

1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048 and 4096.

The default value is 128.

## **DI821**

### **ACTIVE 01 through 08**

If the channel is to be active enter a **YES**.  
If the channel will not be used enter a **NO**.

The default value is YES.

### **AUTO DOWNLOAD**

To automatically download the configuration to the I/O module enter a YES. In the future, if the configuration is to be loaded locally enter a NO.

Default value is YES.

**CLUSTER**

Enter the I/O device cluster of the station where this module is located. The cluster connected directly to the FCI is number zero (0). The number in this field is 0 -7.

Default value is 0.

**FILTER**

Enter the input filter time in msec.

The valid values are:

2, 4, 8 and 16.

The default value is 2.

**LABEL**

Enter a characteristic name to identify the DI821 Module of the station. The name may contain up to 20 characters.

**MONITOR POWER**

To monitor process power source enter a **YES**;  
if not needed enter a **NO**.

Default Value is NO.

**POSITION**

Enter a number from 1 to 12 representing the position of the I/O module relative to the FCI. The I/O module directly next to the FCI is number 1.

Default value is 1.

**REPORT FAULTS 01 through 08**

If channel faults are to be reported, enter a **YES**;  
if not enter a **NO**.

The default value is YES.

**RESERVED OPTIONS**

This field reserved for future use. A zero (0) should be entered as a default value.

**VALUE CYCLE TIME**

Enter the channel value update period (msec) over the fieldbus. Refer to *S800 I/O User's Guide* for details of AF 100 loading in relation to cycle time.

The valid values are:

1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048 and 4096.

The default value is 128.

**DO810**

The DO810 templet is also used for the DO890 except as noted in the field entry descriptions. The DO890 also supports only 4 channels.

**ACTIVE 01 through 16**

If the channel is to be active enter a **YES**.

If the channel will not be used enter a **NO**.

The default value is YES.

**AUTO DOWNLOAD**

To automatically download the configuration to the I/O module enter a YES. In the future, if the configuration is to be loaded locally enter a NO.

Default value is YES.

**CLUSTER**

Enter the I/O device cluster of the station where this module is located. The cluster connected directly to the FCI is number zero (0). The number in this field is 0 -7.

Default value is 0.

**DQ CYCLE TIME**

Enter the channel data quality update period (msec) over the fieldbus. Refer to the *S800 I/O User's Guide* for details of AF 100 loading in relation to cycle time.

The valid values are:

1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, and 4096

The default value is 128.

**LABEL**

Enter a characteristic name to identify the DO810 or DO890 Module of the station. The name may contain up to 20 characters.



For the DO890 module, enter a description that includes the module type for future reference to help alleviate confusion with an DO890 being used in the DO810 templet.

**MONITOR POWER**

To monitor process power source enter a **YES**; if not needed enter a **NO**. DO890 with any channel supervision requires that **MONITOR POWER = YES**.

Default Value is NO.

**OSP HOLD 01 through 16**

If an OSP TIMEOUT has occurred, will the DO810 hold the last value during OSP? Enter a **YES** if it will, or enter a **NO** if an OSP VALUE will be used.

The Default is NO.

**OSP TIMEOUT**

Enter the time (msec) after communications is lost, that outputs go to OSP value.

The valid values are:

OFF, 256, 512, and 1024.

The default value is 1024.



**OSP VALUE 01 through 16**

Enter the channel value (0 or 1) to output after an OSP TIMEOUT. If the OSP HOLD has been selected as the OSP value then this field entry is ignored.

The valid values are:

0 or 1.

Where 0 = off and 1 = on.

The default is 0.

**POSITION**

Enter a number from 1 to 12 representing the position of the I/O module relative to the FCI. The I/O module directly next to the FCI is number 1.

Default value is 1.

**REPORT FAULTS 01 through 16**

If channel faults are to be reported, enter a **YES**;  
if not enter a **NO**.

The default value is YES.

**RESERVED OPTIONS**

Used to select I/O module type:

Enter:

0 (zero) for the DO810 Module

542769152 for the DO890 Module (decimal value for all channels without supervision) see table below where a "1" sets supervision for channels 4 - 1.

The default value is 0 (zero).

HEX Bytes	Byte 0 Option Codes	Byte 2 Module Type No.	Byte 1 Channels 16 - 9 (bits)								Byte 0 Channels 8 - 1 (bits)							
Supervised Channels	b7 - b0	b7 - b0	b7	b6	b5	b4	b3	b2	b1	b0	b7	b6	b5	b4	b3	b2	b1	b0
None	\$20	\$5A	1								1							
All	\$20	\$5A	Channels not used - (set = 00 (zeros))								Channels not used - (set = 0 (zero))							
4 and 1	\$20	\$5A																
3 and 2	\$20	\$5A																
2, 1	\$20	\$5A																
etc.	\$20	\$5A																
	\$20	\$5A																

Examples from above:

**All with No Supervision**

hex number = **205A0000**

converted to decimal = **542769152**

**All with Supervision**

hex number = **205A000F**

converted to decimal = **542769167**

**Channels 4 and 1 with Supervision**

hex number = **205A0009**

converted to decimal = **542769161**

**Channels 3 and 2 with Supervision**

hex number = **205A0006**

converted to decimal = **542769158**

**Channels 2 and 1 with Supervision**

hex number = **205A0003**

converted to decimal = **542769155**

**VALUE CYCLE TIME**

Enter the channel value update period (msec) over the fieldbus. Refer to *S800 I/O User's Guide* for details of AF 100 loading in relation to cycle time.

The valid values are:

1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048 and 4096.

The default value is 128.

## DO814

### **ACTIVE 01 through 16**

If the channel is to be active enter a **YES**.  
If the channel will not be used enter a **NO**.

The default value is YES.

### **AUTO DOWNLOAD**

To automatically download the configuration to the I/O module enter a YES. In the future, if the configuration is to be loaded locally enter a NO.

Default value is YES.

### **CLUSTER**

Enter the I/O device cluster of the station where this module is located. The cluster connected directly to the FCI is number zero (0). The number in this field is 0 -7.

Default value is 0.

### **DQ CYCLE TIME**

Enter the channel data quality update period (msec) over the fieldbus. Refer to the *S800 I/O User's Guide* for details of AF 100 loading in relation to cycle time.

The valid values are:

1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, and 4096

The default value is 128.

### **LABEL**

Enter a characteristic name to identify the DO814 Module of the station. The name may contain up to 20 characters.

**MONITOR POWER**

To monitor process power source enter a **YES**; if not needed enter a **NO**.

Default Value is NO.

**OSP HOLD 01 through 16**

If an OSP TIMEOUT has occurred, will the DO814 hold the last value during OSP?  
Enter a **YES** if it will, or enter a **NO** if an OSP VALUE will be used.

The Default is NO.

**OSP TIMEOUT**

Enter the time (msec) after communications is lost, that outputs go to OSP value.

The valid values are:

OFF, 256, 512, and 1024.

The default value is 1024.

**OSP VALUE 01 through 16**

Enter the channel value (0 or 1) to output after an OSP TIMEOUT. If the OSP HOLD has been selected as the OSP value then this field entry is ignored.

The valid values are:

0 or 1.

Where 0 = off and 1 = on.

The default is 0.

**POSITION**

Enter a number from 1 to 12 representing the position of the I/O module relative to the FCI. The I/O module directly next to the FCI is number 1.

Default value is 1.

**REPORT FAULTS 01 through 16**

If channel faults are to be reported, enter a **YES**;  
if not enter a **NO**.

The default value is YES.

**RESERVED OPTIONS**

This field reserved for future use. A zero (0) should be entered as a default value.

**VALUE CYCLE TIME**

Enter the channel value update period (msec) over the fieldbus. Refer to *S800 I/O User's Guide* for details of AF 100 loading in relation to cycle time.

The valid values are:

1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048 and 4096.

The default value is 128.

**DO820****ACTIVE 01 through 08**

If the channel is to be active enter a **YES**.  
If the channel will not be used enter a **NO**.

The default value is YES.

**AUTO DOWNLOAD**

To automatically download the configuration to the I/O module enter a **YES**. In the future, if the configuration is to be loaded locally enter a **NO**.

Default value is YES.

**CLUSTER**

Enter the I/O device cluster of the station where this module is located. The cluster connected directly to the FCI is number zero (0). The number in this field is 0 -7.

Default value is 0.

### **DQ CYCLE TIME**

Enter the channel data quality update period (msec) over the fieldbus. Refer to the *S800 I/O User's Guide* for details of AF 100 loading in relation to cycle time.

The valid values are:

1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, and 4096

The default value is 128.

### **LABEL**

Enter a characteristic name to identify the DO820 Module of the station. The name may contain up to 20 characters.

### **MONITOR POWER**

To monitor process power source enter a **YES**;  
if not needed enter a **NO**.

Default Value is NO.

### **OSP HOLD 01 through 08**

If an OSP TIMEOUT has occurred, will the DO820 hold the last value during OSP?  
Enter a **YES** if it will, or enter a **NO** if an OSP VALUE will be used.

The Default is NO.

### **OSP TIMEOUT**

Enter the time (msec) after communications is lost, that outputs go to OSP value.

The valid values are:

OFF, 256, 512, and 1024.

The default value is 1024.

**OSP VALUE 01 through 08**

Enter the channel value (0 or 1) to output after an OSP TIMEOUT. If the OSP HOLD has been selected as the OSP value then this field entry is ignored.

The valid values are:

0 or 1.

Where 0 = off and 1 = on.

The default is 0.

**POSITION**

Enter a number from 1 to 12 representing the position of the I/O module relative to the FCI. The I/O module directly next to the FCI is number 1.

Default value is 1.

**REPORT FAULTS 01 through 08**

If channel faults are to be reported, enter a **YES**;  
if not enter a **NO**.

The default value is YES.

**RESERVED OPTIONS**

This field reserved for future use. A zero (0) should be entered as a default value.

**VALUE CYCLE TIME**

Enter the channel value update period (msec) over the fieldbus. Refer to *S800 I/O User's Guide* for details of AF 100 loading in relation to cycle time.

The valid values are:

1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048 and 4096.

The default value is 128.

## Special Applications Runtime Templates

The runtime templates for these FCMs are implemented only in systems having special applications software installed.

### **SP\_ALGO1\_FCM**

The runtime templet for this FCM is implemented only in systems having special application software installed.

### **SP\_ALGO2\_FCM**

The runtime templet for this FCM is implemented only in systems having special application software installed.

### **SP\_ALGO3\_FCM**

The runtime templet for this FCM is implemented only in systems having special application software installed.

### **SP\_ALGO4\_FCM**

The runtime templet for this FCM is implemented only in systems having special application software installed.

### **SP\_ALGO5\_FCM**

The runtime templet for this FCM is implemented only in systems having special application software installed.



## BRKPTS

The Breakpoint Sets (BRKPTS) object is a child of CCF object. One BRKPTS object is required for each breakpoint set being defined for the system. See [Runtime Version of Breakpoint Sets Template](#) on page 62.

### NAME OF SET

The NAME OF SET field specifies the name of the linearization table used when referencing the breakpoint set.

Valid entry is a string of up to 4 ASCII characters.

Default is: blank spaces

### X VALUE

The X VALUE field specifies the x value for one of up to 11 x, y pairs for the breakpoint table.

Valid entry is a floating point number.

### Y VALUE

The Y VALUE field specifies the y value for one of up to 11 x, y pairs for the breakpoint table.

Valid entry is a floating point number.



**A**

Abnormal State 45, 50, 214  
Abnormal State Change 65  
About 29  
AC410 Subsystem Status Display 130  
AC460 Subsystem Status Display 125  
Acknowledge 67, 212 to 213  
Adaptive Mode 57  
Address (Device Number) 123  
AdvaBuild 19  
Advant OCS 19  
AF100 Interface Status Area 149  
AF100 Interface Status Menu 150  
AI810 286  
AI820 288  
AI830 291  
AI835 294  
AI890 286 to 287  
Alarm  
    Acknowledgement 67  
    Check 64  
    Checking 58  
    Detection 64  
    Indication 65  
    Indications 46  
    Limit Parameters 58  
    Limits 66  
    Post 38, 64  
    Posting 58  
    Priority 65  
Alarm Acknowledge 212 to 213  
Alarm Buttons 50  
Alarm Check 38, 42  
Alarm Configuration 38, 42  
Alarm Post 42

Analog Input 288  
Analog Output 300  
Analog output 297  
AO810 templet 297  
AO820 300  
AO890 297 to 298  
Archive Messages 140  
Area Alarm Display, CCF 77  
Area Graphic Display, CCF 78  
Area Status Display, CCF 75  
Aspect Browser 31  
Auto Expand Trend Y-Axis 53  
Auto/Manual Transfer, Transfer, Auto/Manual 205  
Automatic Mode 48  
Automatic Reset Balancing 206

**B**

Balance Mode 240  
Base Gain 233  
Base Rate 58  
Base Reset 234  
BASERATE 216  
Baserate 145  
Bias 49  
Bias Mode 240  
Bipolar Analog Output 300  
Bit Pattern, TLL File 117  
Breakpoint Sets Templet 61, 262 to 321  
    NAME OF SET field 321  
    X VALUE field 321  
    Y VALUE field 321  
BRKPTS, Breakpoint Sets Templet 321  
Browser, Aspect 31  
Bumpless Transfer, Setpoint 202

**C**

Cascade Control System 209  
CCF 19  
CCF Load 144  
CCF Runtime Displays 37  
Change User 30  
CI520/CI522 Submodule Status 148  
CI541 Submodule Status 170  
CI810A FCI 154  
CI820 FCIs 154  
Cluster, S800 I/O 152  
Comp Mode Restrict 221  
Computer Mode 48, 69, 72  
Configurable Control Functions (CCF) 37  
Confirm, Faceplate Button 47  
Connectivity Server 19  
Continuous Loops 44  
Control Action Type, Integral 234  
Control Limit Parameters 58  
Control Output Tracking 49  
Control Packages Overview 31  
Control Structure 198  
Controller Action 234  
Controller Subsystem Status Display 131, 193  
Counter Display, TLL 110  
CPU Load 144  
Cutout State 217

**D**

Data Entry FCM 145, 280  
Database Summary Displays, TLL 119  
DCN 19  
Deviation  
    Bad Quality 58  
    Control Limit 58  
    Deadband 38, 58  
Device  
    Alarms 65  
    Loop Faceplate 43 to 44  
    Loops 42

    Number (Address) 123  
    Parameter 58  
Device Descriptor State 43  
Device Loop Templet 223  
DI810 303  
DI814 templet 306  
DI890 303  
Diagnostic (Status) Displays 121  
Diagnostic Message List 121  
Differential Analog Input 288  
Direct I/O Display 193  
Display Access 31  
Display Request 181  
Display Request Button 132  
Displays, CCF 37  
    Area Alarm 77  
    Area Graphic 78  
    Area Status 75  
    Device Faceplate 43 to 44  
    Group Alarm 74  
    Group Graphic 73  
    Group Status 68  
    Group Trend 71  
    Loop Detail 37  
    Loop Faceplate 40 to 41, 44  
    Loop FCM 51, 59  
Displays, Diagnostic  
    AC410 Subsystem Status 130  
    AC460 Subsystem Status 125  
    Controller Subsystem Status 131, 193  
    Diagnostic Messages 139  
    System Performance 144  
    System Status 122  
    Turbo Node Subsystem Status 137  
    Turbo/Console Node Subsystem Status 137  
Displays, I/O  
    Direct I/O 193  
    PROFIBUS Device 174  
    PROFIBUS LAN 170  
    PROFIBUS Module/Channel 178

S100 Device Status 165  
 S100 LAN 162  
 S800 Device 156  
 S800 LAN 148  
 S800 Station 152  
 TRIO Block 188  
 TRIO LAN 182  
 Displays, Overview (Graphic) 32  
 Displays, TCL 79  
   Message 80  
   Recipe Detail 98  
   Sequence Debug 93  
   Sequence Detail 87  
   SFC 89  
   Unit Detail 83  
   Unit Overview 82  
 Displays, TLL 103  
   Counter 110  
   Database Summary 119  
   File 117  
   I/O Point 115  
   Register 108  
   Segment 103  
   Sequencer 118  
   Timer 112  
 Displays, Tuning  
   Linearization Update 61 to 62  
   Loop FCM 60  
 DO810 templet 311  
 DO890 311  
 Documentation, On-line 23

**E**

Enable Access Restriction 43  
 Enable Posting 217  
 Eng Deadband 58  
 Engineering Deadband 38  
 Error Squared Option 237

**F**

Faceplate  
   A/M Ratio/Bias Loop 41  
   Alarm Indications 46  
   Continuous Loop 44  
   Device Loop 43 to 44  
   Operator Controls 47  
   PID Loop 40  
   Process Bargraphs and Values 46  
   Reduced Size 39  
   Tag Identification 45  
   TLL Counter 111  
   TLL I/O Point 116  
   TLL Program Elements 106  
   TLL Register 109  
   TLL Timer 114  
   Tool Tips 50  
 FCI Module 152  
 FCI Status 154  
 FCM 20  
   Loop Display 51  
   Mode Status 60  
   Output Mode Status 60  
   Setpoint Mode Status 60  
   Setpoint Value 60  
 FCM TMPL (FCM Templet) Button 51, 60  
 FeedBack 49  
 FeedForward 49  
 Feedforward Control 203  
 FeedForward-FeedBack 49  
 FF/FB Mode 237  
 Field State 42, 58  
 File Display, TLL 117  
 Filled Plot 99  
 Find, Sequence Debug 93, 95

**G**

Gain 38, 57, 233  
 Gain Limit 237  
 Goto, Sequence Debug 93, 95

Group Alarm Display, CCF 74  
Group Graphic Display, CCF 73  
Group Status Display, CCF 68  
Group Trend Display, CCF 71

## H

Hard Reset 125  
Help 26  
Hybrid Line Array 99  
Hybrid Plot 99

## I

I/O Point Display, TLL 115  
Illegal State or Command 65  
Inc Minimum Output 242  
Infinite Values 46  
Input  
    Bad Quality 58  
    Hi Alarm Limit 58  
    HiHi Alarm Limit 58  
    Lo Alarm Limit 58  
    LoLo Alarm Limit 58  
    Rate of Change Alarm Limit 58  
Integral Control Action 234  
Integral Type 234  
Interactive Form 237

## L

Limit Parameters  
    Alarm 58, 66  
    Control 58  
Limiting 66  
Linear Table 61  
Linearization Update Display 61  
Link Time-out 240  
Local Mode 48  
Lock State 42, 58  
Log Over 30  
Logon Name and Password 30  
Loop Definition Templet 60, 215

Abnormal State 221  
Bad Alarm Priority 219  
Bad Measure Alarms? 219  
Comp Mode Restrict 221  
Cutout State 217  
Dmnd Processing Mode 222  
Enable Hi Alarms? 218  
Enable Hihi Alarms? 220  
Enable Lo Alarms? 218  
Enable Lolo Alarms? 220  
Enable Posting 217  
Enable Rate Alarms? 219  
Engu Alarm Deadband 219  
Hi Eng. Unit Limit 217  
Hi Limit 218  
High Alarm Priority 218  
Hihi Alarm Priority 220  
Hihi Limit 220  
Lo Alarm Priority 218  
Lo Eng. Unit Limit 217  
Lo Limit 218  
Lolo Alarm Priority 221  
Lolo Limit 220  
Loop Descriptor 215, 223  
Loop State 217  
Loops To Process Edit Window 222  
Measurement Units 217  
Processing Phase 216  
Processing Rate 216  
Rate Alarm Priority 219  
Rate Limit 219  
Suppress Alarms 217  
Trend Rate 216  
Unit ID 221  
Loop Descriptor 215, 223  
Loop Detail Display 198  
    Continuous Loops 38  
    Device Loop 42  
    Tuning Parameters 38  
Loop Detail Display, CCF 37

Loop Faceplate, CCF 40 to 41, 44  
 Loop FCM Display 51, 59, 198  
 Loop FCM Templet 60  
 Loop Mode 38, 42, 58  
 Loop Parameter 58  
 Loop Setup 38, 42  
 Loop State 217  
 LOOP TMPL (Loop Definition Templet) 51, 60  
 Loop Tuning 38  
 Loop, Control 54  
 Loop, Tuning 57

## M

Manual Enable 43  
 Manual Mode 48  
 Manual Reset 206  
 Manual Reset Mode 236  
 Master Controller 209  
 Measured Variable Alarm Limit 58  
 Measurement Units 217  
 Memory Utilization 145  
 Message Display, Diagnostic 139  
 Message Display, TCL 80  
 Messages, Archive 140  
 Microscan Algorithm 234  
 MOD 300 Objects 33

## N

NAME OF SET field  
     Breakpoint Sets Templet 321  
 Node 20

## O

Object Browser 198  
 Objects, MOD 300 33  
 On-line Documentation 23  
 On-Line Help 26  
 Operator Controls 47  
 Operator Workplace 29  
 Output 69, 72

Bad Quality 58  
 Control Limit 58  
 Deadband 38, 58  
 Mode 60  
 Rate of Change Alarm Limit 58  
 Tracking 49  
 Value 39, 51  
 Output Tracking 206  
 Override Mode 42, 58  
 Overview Type Displays 32

## P

Page Selector Alarm Panel 32  
 Phase 38, 42, 58  
 Plant Explorer 29  
 Plot Display, TCL Array 99  
 Plot, Filled 99  
 Preact 38, 57  
 Preact Time 234  
 Print Active Window 36  
 Print Diagnostic Messages 143  
 Print to file 143  
 Process Bargraphs and Values 46  
 Processing Phase 145, 216  
 Processing Rate 58, 145, 216, 233  
 PROFIBUS Device Display 174  
 PROFIBUS LAN Display 170  
 PROFIBUS Module/Channel Display 178  
 Proportional Response 233

## Q

Quality Parameters  
     DeviationBad 58  
     InputBad 58  
     OutputBad 58  
     SetpointBad 58

## R

Ramp 50  
 Ratio 49

Ratio Mode 241  
Ratio Or Bias Balancing 206  
Recipe  
    Changing Values 99  
    Detail Display, TCL 98  
    Loading 85  
Reduced Faceplates 39  
Redundancy Commands, TRIO 184  
Redundancy Request 131  
Redundancy Status, AF 100 150  
Redundancy Status, TRIO LAN 184  
Redundant Device, S100 162  
Refresh Rate, Trend 52  
Register Display, TLL 108  
Remote Mode 48  
Reset 38, 57, 234  
Reset CCF 145  
Reset CPU 145  
Reset for Upgrade 125  
Restore Trace (TCL Array Plot) 101  
Restricted Parameters (Computer Mode) 48  
Revert User 30  
RTD Input 291  
Ruler (TCL Array Plot) 101  
Rungs, Segment display 104

## S

S100 Device Status Display 165  
S100 LAN Display 162  
S100 Smoothstart Start-up Sequence 166  
S100 Warmstart 167  
S800 Device Display 156  
S800 LAN Display 148  
S800 Station Display 152  
Scan Rate 38, 42, 58, 216, 223  
Screen Dump 143  
Security Level 30  
Segment Display, TLL 103  
Sequence  
    Debug 93

    Loading onto a Unit 84  
    Mode 86  
    Removing 86  
    State 85  
    Status 86  
    Steps 88  
Sequence Debug Display  
    Breakpoint 96  
    Trace 95  
Sequence Debug Display, TCL 93  
Sequence Detail Display, TCL 87  
Sequence List 88, 94, 118  
Sequence Source 94  
Sequencer Display, TLL 118  
Server Status 35  
Setpoint 69, 72  
    Bad Quality 58  
    Hi Control Limit 58  
    Lo Control Limit 58  
    Mode 60  
    Tracking Mode 49  
Setpoint Value 39, 51, 60  
SFC (Sequential Function Chart) Display 89  
Simulation Mode 42, 58  
Slave Controller 209  
State Alarm 65  
State Definitions, System 124  
States, FCM Status 60  
Status Displays 121  
Subsystem Name 123  
Subsystem Status 123  
Suppress Alarms 217  
Switch Bus 184, 187  
Switchover 125  
System Performance Display 144  
System Status Display 122

## T

Tag  
    Descriptor 45



- Identification 45
  - Name 45
  - Taylor Control Language (TCL) Displays 79
  - Taylor Ladder Logic 103
  - TCL Array Plot Symbol 99
  - TCL Editor 79
  - TCL Message Display 80
  - TCL Recipe Editor 79
  - TCL Reply 81
  - TCL Sequences 79
  - TCL Version Mismatch 80
  - Templet
    - Breakpoint Sets 321
  - Templets
    - Device Loop 63, 223
    - FCM 37
    - Loop Definition 37
  - Thermocouple Input 294
  - Timeout 65
  - Timer Display, TLL 112
  - TLL 103
    - Device 105
    - I/O Point Faceplate 107
    - Load Segment 105
    - Remove Segment 106
  - Tool Tips, Faceplate 50
  - Trace Steps 95
  - Trace Variables 95
  - Trace Visibility, (CCF) Trend 53
  - Transition (Unknown State or Timeout) 65
  - Trend
    - Auto Expand Y-Axis 53
    - Data Source 52
    - Modifying Y-axis 53
    - Refresh Rate 38, 52
    - Ruler Dialog 54
    - Trace Visibility 53
    - Use Output Limits for Y-Axis 53
  - Trend Rate 38, 58, 69, 216
  - TRIO Block Display 188
  - TRIO LAN Display 182
  - Tuning
    - Breakpoint Sets Templet 62
    - Changing Parameter 57
    - FCM Templet 59, 62
    - Loop Definition Templet 59, 62
    - Loop Detail Display 57
    - Loop FCM Display 57, 59
    - Parameters 38
  - Turbo Node Subsystem Status Display 137
  - Turbo/Console Node Subsystem Status Display 137
- U**
- Unacknowledged Alarm 46, 65
  - Unit Detail Display, TCL 83
  - Unit Overview Display, TCL 82
  - Unknown State 65
  - Use Output Limits Y-Axis 53
  - User
    - Logon 30
    - Manager 30
    - Rights 30
- W**
- Workplace Login 29
- X**
- X VALUE field
    - Breakpoint Sets Templet 321
- Y**
- Y VALUE field
    - Breakpoint Sets Templet 321







3BUR002418R4101 Litho in U.S.A. May 2005  
Copyright © 2005 by ABB. All Rights Reserved  
® Registered Trademark of ABB.  
™ Trademark of ABB.

<http://www.abb.com/control>

Automation Technology Products  
Wickliffe, Ohio, USA  
[www.abb.com/processautomation](http://www.abb.com/processautomation)  
email: [industrialitsolutions@us.abb.com](mailto:industrialitsolutions@us.abb.com)

Automation Technology Products  
Västerås, Sweden  
[www.abb.com/processautomation](http://www.abb.com/processautomation)  
email: [processautomation@se.abb.com](mailto:processautomation@se.abb.com)

Automation Technology Products  
Mannheim, Germany  
[www.abb.de/processautomation](http://www.abb.de/processautomation)  
email: [marketing.control-products@de.abb.com](mailto:marketing.control-products@de.abb.com)