Honeywell

Experion

Control Building

User's Guide

EP-DCXX64

R300.1

5/06

Notices and Trademarks

Copyright 2006 by Honeywell International Inc.
Release 300.1 **May 5, 2006**

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

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

Honeywell, PlantScape, Experion, and **TotalPlant** are registered trademarks of Honeywell International Inc.

Other brand or product names are trademarks of their respective owners.

Honeywell International

Process Solutions

2500 West Union Hills

Phoenix, AZ 85027

1-800 343-0228

About This Document

The procedures in this guide are intended to give you the ability to perform basic tasks within the Control Builder application such as configuring hardware devices, continuous control strategies, and sequential control strategies. Only representative forms are shown to illustrate a procedure/concept.

Release Information

Document Name	Document ID	Release Number	Publication Date
Control Building User's Guide - cb_g	EP-DCXX64	300.1	5/06

References

The following list identifies all documents that may be sources of reference for material discussed in this publication.

Document Title

Contacts

World Wide Web

The following Honeywell web sites may be of interest to Process Solutions customers.

Honeywell Organization	WWW Address (URL)
Corporate	*http://www.honeywell.com*
Honeywell Process Solutions	*http://hpsweb.honeywell.com*

Telephone

Contact us by telephone at the numbers listed below.

Location	Organization	Phone
United States and Canada	Honeywell IAC Solution Support Center	1-800-822-7673
Europe	Honeywell TAC-EMEA	+32-2-728-2704
Pacific	Honeywell Global TAC - Pacific	1300-300-4822 (toll free within Australia) +61-8-9362-9559 (outside Australia)
India	Honeywell Global TAC - India	+91-20-2682-2458
Korea	Honeywell Global TAC - Korea	+82-2-799-6317
People's Republic of China	Honeywell Global TAC - China	+86-10-8458-3280 ext. 361
Singapore	Honeywell Global TAC - South East Asia	+65-6580-3500
Taiwan	Honeywell Global TAC - Taiwan	+886-7-323-5900
Japan	Honeywell Global TAC - Japan	+81-3-5440-1303
Elsewhere	Call your nearest Honeywell office.	

Symbol Definitions

The following table lists those symbols used in this document to denote certain conditions.

Symbol	Definition
	**ATTENTION:** Identifies information that requires special consideration.
	**TIP:** Identifies advice or hints for the user, often in terms of performing a task.
	**REFERENCE -EXTERNAL:** Identifies an additional source of information outside of the bookset.
	**REFERENCE - INTERNAL:** Identifies an additional source of information within the bookset.
CAUTION	Indicates a situation which, if not avoided, may result in equipment or work (data) on the system being damaged or lost, or may result in the inability to properly operate the process.
	**CAUTION**: Indicates a potentially hazardous situation which, if not avoided, may result in minor or moderate injury. It may also be used to alert against unsafe practices. **CAUTION** symbol on the equipment refers the user to the product manual for additional information. The symbol appears next to required information in the manual.
	**WARNING**: Indicates a potentially hazardous situation, which, if not avoided, could result in serious injury or death. **WARNING** symbol on the equipment refers the user to the product manual for additional information. The symbol appears next to required information in the manual.

Symbol	Definition
	**WARNING, Risk of electrical shock**: Potential shock hazard where HAZARDOUS LIVE voltages greater than 30 Vrms, 42.4 Vpeak, or 60 VDC may be accessible.
	**ESD HAZARD:** Danger of an electro-static discharge to which equipment may be sensitive. Observe precautions for handling electrostatic sensitive devices.
	**Protective Earth (PE) terminal**: Provided for connection of the protective earth (green or green/yellow) supply system conductor.
	**Functional earth terminal**: Used for non-safety purposes such as noise immunity improvement. NOTE: This connection shall be bonded to Protective Earth at the source of supply in accordance with national local electrical code requirements.
	**Earth Ground**: **Functional earth connection.** NOTE: This connection shall be bonded to Protective Earth at the source of supply in accordance with national and local electrical code requirements.
	**Chassis Ground**: Identifies a connection to the chassis or frame of the equipment shall be bonded to Protective Earth at the source of supply in accordance with national and local electrical code requirements.

Contents

WHAT DO YOU WANT TO DO? 1

CONTROL BUILDER PURPOSE 3

Getting Started 3
Basic concepts and terms 3
Conventions 4

What is Control Builder? 5
Control Builder application 5

What is a control strategy? 5
Maximum number of Control Builder instances 5

Configuring a control strategy 6
Creating hardware and control module instances 6
Creating Sequential Control Module instances 8
Creating Custom Algorithm Blocks and Custom Data Blocks instances 8

I/O module and controller compatibility 9
Determining I/O module support 9

CONTROL BUILDER CONFIGURATION 13

Starting Control Builder 13
Launching from Configuration Studio to access Control Builder 13
Login to Server 15
Configuring Windows user account in Station 16

Layout of the main window 19
Menus 20
Toolbar 24
Control Drawing 27
Status Line 27
Opening and navigating a tree window 28
Changing the state of a tree window 31
Moving a docked tree window 33
Closing a tree window 33

Block and configuration form tabs tables 34
SYSTEM 34
AUXILIARY 36
DEVCTL 37

DATAACQ ... 38
HIWAY ... 39
HIWAYIF ... 41
IOMODULE ... 42
IOCHANNEL ... 44
LIOM_FB ... 45
LIOM_IOC ... 47
LIOM_IOM ... 48
PMIO ... 50
LOGIC ... 53
REGCTL ... 55
SCM ... 59
SERIES_C_IO ... 60
UCNIF ... 61
UTILITY ... 62
FIELDBUS ... 63
HARTIO ... 64
DNETIF ... 65
AB_DRIVE_IF ... 67
AGA ... 69
PBUSIF ... 70
RAILIO ... 77
FBUSIF ... 78
EXCHANGE ... 80
PULSEINPUT ... 81
RAIL_IO_HAZ ... 83
QIMPACT ... 84
JAGXTREME ... 85

Configuring modules ... 86
Defining Series C Function Blocks ... 87
Configuring modules - Main tab ... 87
Configuring modules - Module Configuration tab ... 90
Configuring modules - Channel Configuration tab ... 92
Configuring modules - Server History tab ... 94
Configuring modules - Server Displays tab ... 96
Configuring modules - Diagnostic Configuration tab ... 98
Configuring modules - Status/Data tab ... 100
Configuring modules - Version tab ... 101
Configuring modules - Control Confirmation tab ... 103
Configuring modules - Identifcation tab ... 105
Configuring Chassis Analog I/O module ... 107
Configuring Chassis Digital I/O module ... 109
Configuring Chassis Diagnostic I/O Module ... 111
Configuring the Serial Interface Module ... 113
Configuring the Pulse Input Module ... 115

CONTROL BUILDER OPERATIONS.....117

Creating hardware modules 117

Creating a CPM and CEE..... 118
Configuring the Main tab 118
Configuring the Server Displays tab 123

Creating an ACE and CEE..... 129

Creating an External Server..... 139

Creating an Inter Cluster Gateway Block 145
Ancillary activity: REEOUT block and SP push 149

Creating an IOLIM and IOLINK..... 149

Creating Redundancy Modules 155

Creating an instance of Input/Output Modules..... 159

Creating function blocks for Input/Output Modules (IOM) or Processors (IOP)163
Creating Series C I/O AI-HART 164
Creating Series C I/O AI-LLMUX.....165
Creating Series C I/O AO-HART 167
Creating Series C I/O DI-24 168
Creating Series C I/O DI-HV 170
Creating Series C I/O DO-24B 172
Creating PM I/O High Level Analog Input IOP 174
Completing the configuration form.....175
Creating PM I/O HART High Level Analog Input IOP.....182
Creating PM I/O Low Level Analog Multiplexer Input IOP 184
Creating PM I/O Analog Output IOP 186
Creating PM I/O HART Analog Output IOP.....188
Creating PM I/O Digital Input IOP 190
Creating PM I/O Digital Output IOP.....192
Creating PM I/O STIMV IOP 194

Creating a control module 196
Creating and saving a control module 197
Copying control modules.....203
Assigning CMs and IOMs to CEE 204
Assigning IOPs to IOLINK.....213
Creating an instance of a basic function block 219
Assigning PMIO IOC block to IOP.....221
Unassigning PMIO IOC block from IOP 226
Copying a function block 229
Moving function blocks within a chart 233
Deleting function blocks 234
Using the Parameters Configuration form 235
Configuring alarms 241

Requesting value changes for configuration parameters 242

Creating a strategy to use insertion points 244
Creating a CAB with insertion program 244
Creating a control module to include insertion points 245
Configuring insertion points 245
Loading control module with insertion points 247
Activating control module with insertion points 247
Checking insertion point status 248
Deleting insertion points 248

Connecting and disconnecting blocks 249
Connecting blocks with insert wire 249
Disconnecting blocks 252
Repositioning connecting wires using drag and drop 252
Repositioning connecting wires using vertices 255
Connecting blocks with Parameter Connector option 256
Cross references function 258
Enabling cross-references 260
Printing cross-references 261
Using the Point Selection tool 261
Using peer-to-peer communications 264

Associating I/O Channels to I/O Modules (IOMs) 267

Inserting OLE objects into charts 271
Linking an OLE object into a chart 271
Editing a linked file in a chart 273
Embedding an OLE object into a chart 273
Editing an embedded file in a chart 275
Deleting an OLE object from a chart 276

Using Server Scripting in Control Builder 277
Server Scripting overview 277
Using Script Editor 277

Printing from Control Builder 280
Printing features 280
Printing preferences 281
Page breaks 281
Print permissions 282
Page Setup 283
Header Setup 284
Footer Setup 285
Printing Options dialog 287
Printing reports 290
Printing control drawings 290
Printing charts 290

Using the Regulatory Loop Wizard 292

Exporting function block configurations 303
Some general export considerations 303
Opening the Export dialog box 304
Exporting a selected project 305
Exporting selected function block(s) 306
Exporting a particular type of function blocks 308
Canceling an export currently in progress 309
Commencing an Export while a Load is in progress 310
Commencing a Load while an Export is in progress 312
Exporting data to a user selected directory 313

Importing function block configurations 315
Some general import considerations 316
Opening the Import dialog box 317
Importing function blocks from selection list 318
Importing function blocks from directory 322
Canceling an Import currently in progress 324
Commencing an Import while a Load is in progress 325
Commencing a LOAD while an Import is in progress 328

Using Module Hierarchy 329
Purpose 329
Initializing Module Hierarchy 329
Printing the Module Hierarchy chart 331
Module Hierarchy reports 333

Working with Profit Loop PKS 335
What is Profit Loop PKS? 336
Implementing Profit Loop PKS 336
Converting a PID-based control loop to PID-PL 336
Conversion phase 337
Configuration phase 339
Download phase 340
Naming considerations 340
Conversion files 340
Reverting to a PID block 341
Profit Loop PKS Assistant overview 341
Starting the Assistant 342
Starting from the Project tab 343
Starting from the Monitoring tab 343
Starting from an Experion Station Point detail display 344
Working with Profit Loop PKS Assistant displays 345
Obtaining basic information 347
Defining a model with Profit Loop PKS Assistant 348
Define Model by Direct Entry 349
Defining a Model from PID Tuning 350
Defining a Model by Loop Type 351
Defining a Model by Step Testing 358

Download a model ... 363
Reverting to a previous model ... 364
Exporting and importing Model Definitions ... 367
Copying model definitions ... 369
Tune a PID-PL-based controller ... 370
Troubleshooting a PID-PL-based controller ... 372
Controlling access to Profit Loop PKS ... 374

Loading a control strategy ... 375
About load operations ... 376
Loaded versus project database versions ... 376
Load initiation and load dialog box ... 376
Load action with compare parameters function ... 377
Load options for server history and server displays configuration ... 378
Initial load order guidelines ... 378
Component deletion considerations ... 379
Loading control strategy components ... 379
Using Upload command ... 397
Using Upload With Contents command ... 398
Using Compare Parameters ... 400
Using Compare Parameters reports ... 409
Copying control strategies using templates ... 411
Using the Change Parent Template wizard ... 419
Loading only server configuration ... 425

Using Bulk Build utility ... 428
Identifying functions on Create Bulk Build List dialog ... 429
Identifying functions on Set Name and Count dialog ... 432
Identifying functions on Select Parameters dialog ... 434
Creating Bulk Build List file ... 437
Viewing sample spreadsheet file ... 440
Viewing sample Access database file ... 443
Identifying functions on Read Bulk Build File ... 446
Reading Bulk Build File ... 449

On-line monitoring using Control Builder ... 452
Initial activation order guidelines ... 452
Activating the CEE ... 453
Setting the CEE inactive ... 462
Setting I/O active ... 465
Setting I/O inactive ... 467
Changing parameters while monitoring ... 469
Operator actions versus IOP outputs ... 472
Deleting ACE/CEEACE block ... 474
Issuing a shutdown command and using Checkpoint to restore ACE ... 477
Recovering from power failure ... 480

CONTROL BUILDER SYSTEM ADMINISTRATION ... 481

Setting system preferences 481
Enabling selected system functions 481
To establish General properites using System Preferences dialog 482
To establish Pins and Wires properties by using Systems Preferences dialog 485
To establish IP addresses using the Embedded FTE tab on System Preferences dialog 487
To change the Base IP Address updates IP information in ERDB 490

Setting user preferences 490

CONTROL BUILDER TROUBLESHOOTING AND MAINTENANCE 495

Database maintenance 495
Deleting CM and SCM (optional) 495

APPENDIX A - SMART TRANSMITTER INTEGRATION WITH PM I/O 497

Introduction 497

Reviewing the Smart Transmitter Interface IOP 498
About digital integration 498
STI IOP features 499
Reranging 500
Bad database protection 500
Alarming 500
Off-line or on-line configuration 501
Transmitter diagnostics 501
Signal noise immunity 501

Integrating the Smart Transmitter 502
Hardware considerations 502
Bench check 502
Commission loop 503

Reviewing DE communication protocol 503
DE versus analog format 504
Broadcast 4-byte format (PV) 506
Broadcast 6-byte format (PV, SV, and database) 506
Smart device communication architecture 508
Physical layer 508
Data link layer 510
Application layer 510
Floating point format 511
Error checking 511
Noise immunity 511

What are IOP functions 512
IOP and transmitter database 515

Building point displays for ST devices 517

Using IOP cabling and FTA wiring..... 517
FTA wiring 517
Redundant IOPs..... 521
Intrinsically safe barriers..... 524

Field wiring the transmitter 528

Integrating multivariable field devices 528
STIMV IOP 528
FTA wiring rules 528
Configuration rules 530
DECONF parameter..... 531
Field device in analog or 4-byte mode..... 532
DECONF configuration rules..... 532
Download 534
Database discrepancies 534
Upgrading STI IOP to STIMV IOP..... 534
Synch error..... 535

APPENDIX B - HIERARCHY BUILDING..... 537

Hierarchy building overview..... 537

Hierarchy tree views..... 538
Assignment view 538
Containment view..... 538

CM/SCM containment..... 540
Functional description..... 540
Containing CM/SCM..... 540
Uncontaining CM/SCM..... 546
Uncontain blocks with Projected Parameter(s)..... 547
Error scenarios for containment/uncontainment operations 550
Operations of container and contained CM(s)/SCM(s)..... 552
Load/load with contents..... 567
Performance restrictions..... 573

Parameter projection overview 573
Naming and renaming 573
Origin parameter overview 576
Empty origin parameter 576
Assigning / unassigning / reassigning the origin parameter..... 577
Validating the origin parameter..... 578
Changing the value 580

Using the Projected Parameters tab..... 580
Configuration form overview..... 580

Symbol attributes....583
Adding a Projected Parameter....585
Deleting a Projected Parameter....585
Printing Projected Parameters....586

Connecting Projected Parameter....586
Read-only connections....587
Validation....588
Point Picker....588

Using projected parameter for user templates....589
Propagation....589
Scenarios and examples....590

Block operations involving Projected Parameter....591
Loading block with projected connections....591
Update to Project....591

Reviewing Substitute Name dialog box....592
Using the Point Picker....593
Resolve substitute connections....595

Importing/exporting....596
Containment....596
Projected Parameter....596
Strategy Import/Export dialog box....597

APPENDIX C - USER TEMPLATES....601

Library tree overview....601
Derivation View....601
Containment View....602

Using the library tree....603
Switch between Derivation and Containment View....604
Configure Module Parameters (user templates only)....604
Open control drawing for edit (user templates only), allowed for SCM and CM templates....604
Copy (user templates only)....605
Move library (user templates only)....605
Specialize....605
Change parent....605
Instantiate (as a tree operation, this applies to container blocks only)....606
Promote (user function block templates only)....606
Delete (user templates only)....607
Delete library (not functional in this release.)....607
Identification....608
Dependencies....609
Import/export template (user templates only)....611
Move library (not functional in this release.)....611

Project/monitor tree overview **612**
Identification tab 612
Dependencies tab 613

Using control drawings **614**
Opening a control drawing 614
Copying a control drawing 614

APPENDIX D - BULK EDIT PARAMETERS UTILITY **615**

Overview **615**

Using the Bulk Edit Parameter utility **615**
If your system is licensed for QVCS 616

Accessing Create Bulk Edit List **616**
Accessing Create Bulk Edit List dialog 616
Create Bulk Edit List dialog 617
Typical use example 620

Edit list file examples **622**

Read Bulk Edit List **623**
Accessing the Read Bulk Edit List dialog 624
Read Bulk Edit List dialog 624
Reading a Bulk Edit List (a typical use example) 628

APPENDIX E - USER DEFINED SYMBOLS **633**

Introduction **633**

Viewing symbol library **635**

Creating a user defined symbol **637**
Supported graphics formats 637
Image considerations 637
Examples 638

Adding symbol to library **641**

Deleting symbol from user library **642**

Reloading user defined symbol **642**

Exporting symbol **643**

Assigning symbol to block **644**

User defined template considerations **646**

QVCS and UDS considerations **646**

Defining automatic pin and UDS 647

Defining tree displays and UDS 647

Specifying chart header and footer layouts 648
Select element frame 648
Default font information frame 652
Header and footer frames 652
Defined elements display 652

Printing a chart 657

APPENDIX F - CONTROL BUILDER CHECKPOINT REFERENCE ..659

Using Checkpoint to Save and Restore Data 659

Getting Started 661
Checking status of Engineering Tools application 661
To check Engineering Tools status 662

Configuring preferences for Checkpoint function 662
To configure checkpoint preferences 664

Checkpoint preferences procedural illustration reference 665

Configuring operation permissions for Checkpoint functions 666
To configure operation permissions 667

Checkpoint preferences procedural illustration reference 668

Checkpoint disk space maintenance functional considerations 669

Reviewing Checkpoint Functional Aspects 670
Functional Description 670

Checkpoint file characteristics 671
Checkpoint files and files storage overview 672
Checkpoint functional rules and guidelines 674

Control Builder Checkpoint interface summary 677

Control Builder Interface graphical reference 678

Station Detail Display Checkpoint interface summary 680

System Status Display Checkpoint interface summary 680

System Event Checkpoint interface summary 680

Identifying Functions on the Checkpoint Scheduler Dialog 680

Identifying Functions on the Define Task Dialog 684

Scheduling Checkpoint Tasks 688

To create and schedule an automatic checkpoint task 690
About naming tasks 691
Checkpoint scheduler state conditions related to stop and start actions 692
Task overrun condition 693
Deferral and Resumption Of Node Saves 693
Definition and computation of task's elapsed run time 693
Definition and computation of next run time 693
Checkpoint scheduler graphical reference 695
Identifying Functions on the Archive Checkpoint Files Dialog 714
Archiving Checkpoint Files 717
To archive checkpoint files 719
Archive checkpoint graphical reference 720
Checkpoint file attributes 728
Compatibility attribute 729
Validity attribute 729
Entirety Attribute 729
Identifying Functions on the Save Checkpoint Manually Dialog 732
Pre-selection of nodes to be saved 736
Save arbitration on a given Node 737
Execution states for checkpoint save 737
Deferral and Resumption of entity saves during a Manual Save 738
Initiating Manual Checkpoint Save 738
To initiate a manual checkpoint save 739
Save checkpoint manually graphical reference 741
Identifying Functions on the Restore from Checkpoint Dialog 748
Pre-selection of nodes to restore 751
Restore scope selection and actual restore action 752
Restoration file display reference 754
Order of restore checks 754
Execution states for checkpoint restore 757
Child hardware restore checks 758

Checkpoint restore/save arbitration with other database changing operations 759

Restoring From Checkpoint 763
Restoring Custom Algorithm Block (CAB) 764
To initiate restore from checkpoint 765

Restore from checkpoint graphical reference 767

Rebuilding Checkpoint from Monitoring Tab 784
To initiate checkpoint rebuild 785

Using Detail Displays for checkpoint operations 787
Launching checkpoint operations from Detail Displays 788
Viewing parameters on Detail Displays 790

Checkpoint operations through Detail Displays graphical reference 792

Troubleshooting Checkpoint Function 797

Viewing error logs 798

Identifying checkpoint errors logged but not journaled 798

Checking messages journaled as events 801

Checking warnings and errors Logged but not journaled as events 803

Checkpoint operation with On-Process Migration 805
On-process server migration 805
On-process controller migration 806

Fixing common problems 807
Checkpoint file is still marked compatible after change in CAB Type block 807

CONTROL COMPONENT DISPLAY ELEMENT REFERENCE 809

About detail and group displays 809
Considerations 809

Tables

Table 1 Controllers and supported I/O module families.....9
Table 2 Drag and drop operations for containment 544

Figures

What do you want to do?

This document includes information that you can reference for:

Topic	Link
Review general information about Control Builder	Click here
Review control strategy information	Click here
Create a hardware module	Click here
Review block's configuration forms/tabs	Click here
Configure modules	Click here
Create function blocks for IOP	Click here
Set system and user preferences	Click here
Load a control strategy	Click here
Duplicate an existing control strategy (using Bulk Builder)	Click here

Control Builder Purpose

Getting Started

This section includes information that you can reference for:

Topic	Link
Basic concepts and terms	Click here
Conventions	Click here

Basic concepts and terms

The procedures in this guide are intended to give you the ability to perform basic tasks within the Control Builder application such as configuring hardware devices, continuous control strategies, and sequential control strategies. No attempt is made in the guide to explain the functionality of Control Builder, nor the many different options available when completing block configuration forms. Only representative forms are shown to illustrate a procedure/concept. The many different variations of forms and associated fields for every Function Block (FB) are not shown in this guide.



REFERENCE - EXTERNAL

For a detailed description of intended design use of container and basic blocks, refer to the Experion Control Builder Components Theory, and the Experion Control Builder Components Reference, Component Categories and Types

Conventions

- Terms and type representations

The following table summarizes the terms and type representation conventions used in this Guide.

Term/Type Representation	Meaning	Example
Click	Click left mouse button once. (Assumes cursor is positioned on object or selection.)	Click the Browse button.
Double-click	Click left mouse button twice in quick succession. (Assumes cursor is positioned on object or selection.)	Double click the Station icon.
Drag	Press and hold left mouse button while dragging cursor to new screen location and then release the button. (Assumes cursor is positioned on object or selection to be moved.)	Drag the PID function block onto the Control Drawing.
Right-click	Click right mouse button once. (Assumes cursor is positioned on object or selection.)	Right-click the AND function block.
<F1>	Keys to be pressed are shown in angle brackets.	Press <F1> to view the online Help.
<Ctrl>+<C>	Keys to be pressed together are shown with a plus sign.	Press <Ctrl>+<C> to close the window.
File->New	Shows menu selection as menu name followed by menu selection	Click File->New to start new drawing.
>D:\setup.exe<	Data to be keyed in at prompt or in an entry field.	Key in this path location >D:\setup.exe<.

What is Control Builder?

This section includes information that you can reference for:

Topic	Link
Control Builder application	Click here

Control Builder application

Control Builder is a control building software used to configure Experion's controllers, which can handle all possible control requirements — whether for continuous processes, batch processes, discrete operations or machine control needs.

What is a control strategy?

A control strategy is an organized approach to define a specifc process using detailed information to:

- create control modules in an associated controlled environment
- configure function blocks to enable control applications, and
- runs in a control software infrastructure

Maximum number of Control Builder instances

A total of 6 instances of Control Builder and/or Enterprise Model Builder (EMB) can be running on a workstation at once. For example, you may have 3 instances of Control Builder open and 3 instances of EMB on the same client node, or any combination of the two applications that total 6.

Configuring a control strategy

The following tables list the recommended steps that should be followed when developing a control strategy. Each step references a section to see for detailed information. The table assumes that the Experion Server has already been set-up and configured.

Creating hardware and control module instances

The following table references the complete PM I/O family architecture and the creating of the Series C I/O Modules.

For the complete Series C family architecture refer to the following documents:

- C300 Controller User's Guide - Provides planning and designing activities, as well as the installation, operation, and troubleshooting information for the Series C300 Controller.
- Series C I/O User's Guide - Provides planning, configuration, and troubleshooting information for the Series C I/O Modules.
- Series C Fieldbus Interface Module User's Guide - Provides planning and implementation information for the Series C Fieldbus Interface Module.

Step	Action	See
1	Start Control Builder and open two tree windows.	Starting Control Builder and Opening and navigating a tree window
2	Create Hardware Blocks.	Creating a CPM and CEE, Creating an ACE and CEE Creating an External Server Creating an Inter Cluster Gateway Block Creating an IOLIM and IOLINK Creating Redundancy Modules

Step	Action	See
3	Configure chassis I/O Modules	Configuring Chassis Analog I/O module Configuring Chassis Digital I/O module Configuring Chassis Diagnostic I/O Module Configuring the Serial Interface Module Configuring the Pulse Input Module
4	Create Input/Output Module or Processor	Creating Series C I/O AI-HART Creating Series C I/O AI-LLMUX Creating Series C I/O AO-HART Creating Series C I/O DI-24 Creating Series C I/O DI-HV Creating Series C I/O DO-24B Creating PM I/O HART High Level Analog Input IOP Creating PM I/O Low Level Analog Multiplexer Input IOP Creating PM I/O Analog Output IOP Creating PM I/O HART Analog Output IOP Creating PM I/O Digital Input IOP Creating PM I/O Digital Output IOP Creating PM I/O STIMV IOP
5	Create a Control Module (CM).	Creating and Saving a Control Module
6	Add Function Blocks to the CM.	Creating an instance of a basic function block
7	Configure the Function Blocks.	Using the Parameters Configuration form
8	Connect the Function Blocks.	Connecting Blocks with Insert Wire Connecting blocks with Parameter Connector option

Step	Action	See
9	Assign IOMs and CMs to the Control Execution Environment (CEE).	Assigning CMs and IOMs to CEE
10	Associate I/O Channels to I/O Modules.	Associating I/O Channels to I/O Modules (IOMs)
11	Assign I/O Processors (IOPs) to I/O Link (IOLINK)	Assigning IOPs to IOLINK
12	Assign Process Manager I/O, Input/Output Channel (IOC) block to I/O Processor (IOP)	Unassigning PMIO IOC block from IOP
13	Load the Control Strategy.	Loading control strategy components
14	Activate the Control Strategy.	Activating the CEE Setting I/O active

Creating Sequential Control Module instances

Sequential Control Module (SCM) information can be found in its own user guide.



REFERENCE - EXTERNAL

For a detailed description of intended design use of container and basic blocks, refer to the Experion Control Builder Components Theory.

Creating Custom Algorithm Blocks and Custom Data Blocks instances

Custom Algorithm Block (CAB) and Custom Data Block (CDB) information can be found in its own user guide.



REFERENCE – EXTERNAL

For a detailed description of intended design use of container and basic blocks, refer to the Experion Control Builder Components Theory.

I/O module and controller compatibility

This section includes information that you can reference for:

Topic	Link
Determining I/O module support	Click here

Determining I/O module support

The controller type used (C300, C200, or xPM) has a determination on the I/O module family that can be used. Not all I/O modules are supported by all controller types. Use the following table to determine which I/O modules are supported by the listed controller.

Table 1 Controllers and supported I/O module families

I/O type	C300 support	C200 Support	xPM support
SERIES C			
HLAI w/ HART	X		
LLMUX64	X		
AO16 w/ HART	X		
DI24D32	X		
DI32	X		
DO24D32	X		
DO32	X		
Fieldbus	X		
PMIO			
HLAI	X	X	X
HLAI w/ HART	X	X	X
LLAI	X	X	X
LLMUX	X	X	X

RHMUX	X	X	X
STI	X	X	X
STI-MV	X	X	X
AO8	X	X	X
AO16	X	X	X
AO16 w/ HART	X	X	
Serial Device			X
Serial			X
Pulse			X
DI	X	X	X
DI-24V	X	X	X
DISOE	X	X	X
DO16	X	X	X
DO32	X	X	X
Fieldbus			X
CHASSIS - Series A			
HLAI16		X	
HLAI6		X	
HLAI8 w/ HART		X	
LLAI - TC6		X	
LLAI – RTD6		X	
AO6V		X	
AO6MA		X	
AO8		X	
AO8 w/ HART		X	
Pulse (6I2O)	X	X	

DI120A16I		X	
DI220A16I		X	
DI120A8D		X	
DI120A16		X	
DI24D16I		X	
DI24D16D		X	
DI24D32		X	
DO220A16-Relay		X	
DO24D16I		X	
DO24D16D		X	
DO24D32		X	
Serial	X	X	
Fieldbus		X	
Profibus DP	X	X	
DeviceNet	X	X	
RAIL – Series A			
HLAI		X	
DI24D		X	
DI120A		X	
LLAI-TC		X	
LLAI-RTD		X	
AO		X	
DO24D		X	
DO120A		X	
DO-Relay		X	
RAIL – Series H			

HLAI8		X	
LLAI8		X	
AO8		X	
DI16		X	
DO4		X	
DI16		X	

Control Builder Configuration

Starting Control Builder

To properly start Control Builder, you must:

- launch Configuration Studio, and
- successfully login (with the appropriate security priviledges) using established user-accounts.

Launching from Configuration Studio to access Control Builder

Configuration Studio houses the various server-based applications that are used to configure control strategies. Control Builder is now accessed through Configuration Studio.

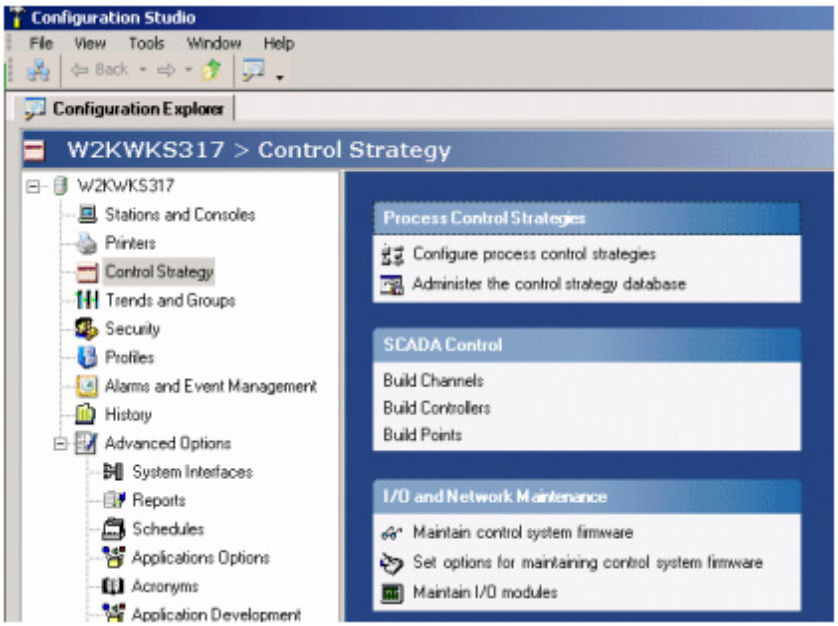


ATTENTION

Refer to the Configuration Studio documentation for information pertaining to its function and features.

The following procedure launches Configuration Studio in order to access Control Builder:

Step	Action	Result
1	Click Start -> Programs -> Honeywell Experion -> Configuration Studio	The Connect dialog box appears.
2	Select a listed server and click the Connect button.	Configuration Studio is launched.

Step	Action	Result
		
3	From the Configuration Explorer tab treeview, click Control Strategy.	The Control Stratgey selections appear.
4	From the Process Control Strategies grouping, click **Configure process control strategies.**	Control Builder launches.
	ATTENTION If the Control Builder icon  appears on the Windows Taskbar at the bottom of screen, the Control Builder program is already running. Click the Control Builder icon to view Control Builder. You can run multiple sessions of Control Builder simultaneously on the same computer, but it is not generally recommended.	
5	Go to the next section *Login to Server*, if user accounts have not been set.	Must login to identify the user, the security level, and the Server that Control Builder is to connect to for this session.

Login to Server

The ability to access Control Builder through Configuration Studio is based on valid login account and security permissions. The security access for login integrates Windows user accounts with Station operator based accounts.



ATTENTION

- A password that is associated with your user name should have been assigned to you. An appropriate server name should also be identified. If necessary, contact your system administrator for the password and server name to be used with your system.
- User name and password come from Station operator-based security or the Windows user account for the computer logon.
- Since Experion Server is providing user authentication, it needs to be running in order to be able to run Control Builder.



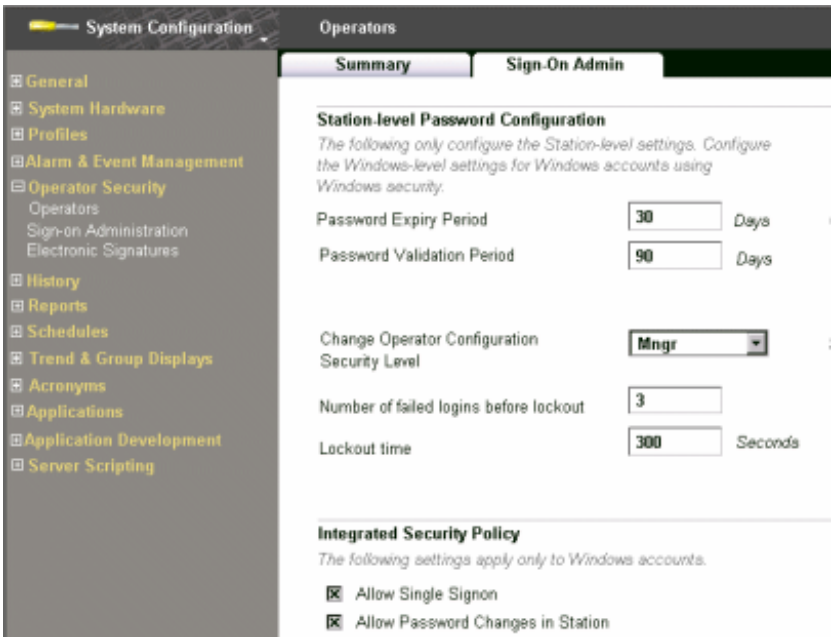
REFERENCE - INTERNAL

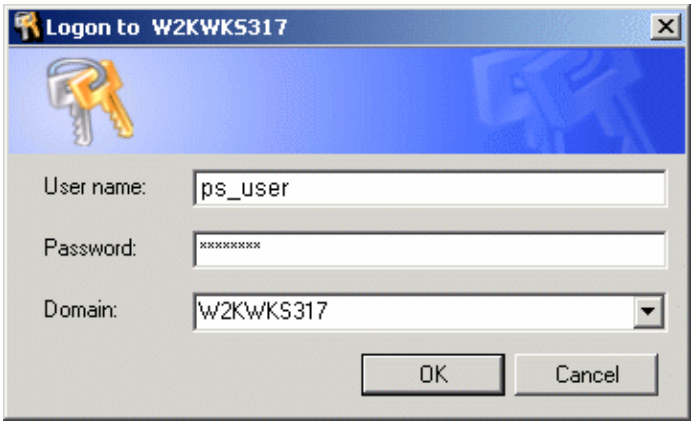
Please review the information in the Configuring Security and Access section of the *Server and Client Configuration Guide* in Knowledge Builder before attempting to login to Control Builder.

Configuring Windows user account in Station

The following procedure sets up a Windows user account through Station.

Step	Action	Result
	**TIP** You must first set up a corresponding Windows user account through Station's Configure->Operator->Operators before you attempt to login to Control Builder using a Windows user account. Be sure the Security Level is set to mngr and Control Level is set to 255. You must have an access level of manager to set up an operator account. Configure the windows-level settings for Windows acccounts using Windows security.	
1	Do you want on enable the Single SignOn function?	• If the answer is Yes, go to Step 2. • If the answer is No, go to Step 4.

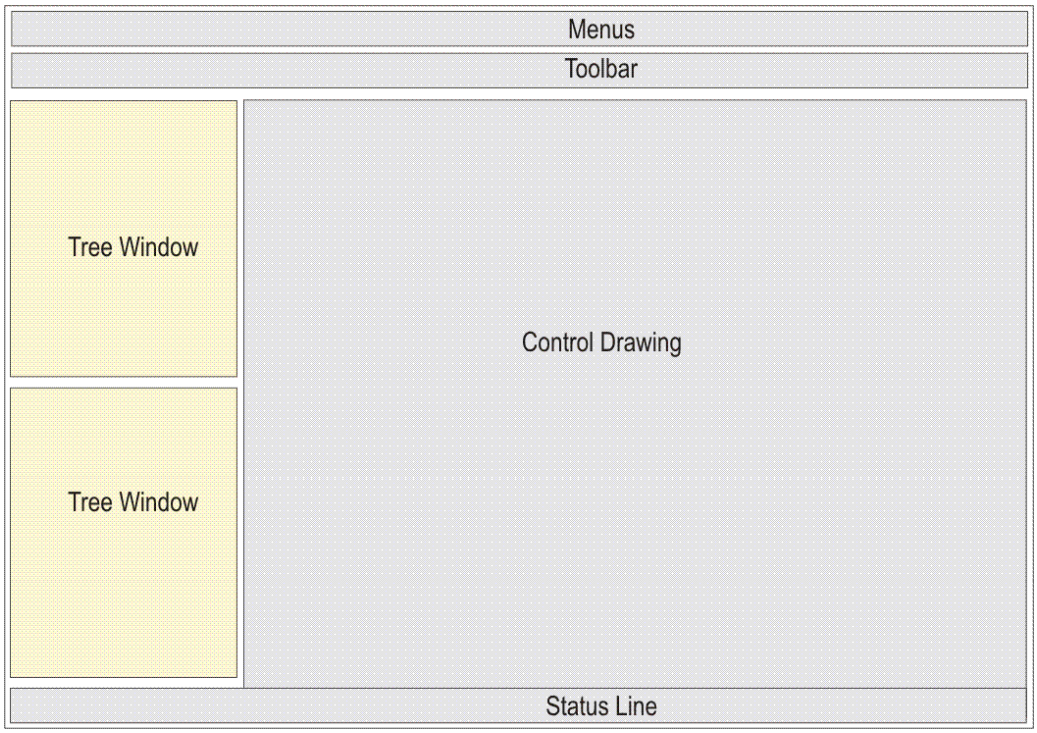
Step	Action	Result
	TIP You must first enable the Allow Single Signon function through Station's Configure->Operators->SignOn Administration configuration form before you can enable this function through Control Builder. You must have an access level of manager to make this change. 	
2	Click the Enable Single SignOn check box.	Check mark appears in the check box, single signon function is enabled, and only the Server Name field is accessible. Login is based on current Windows user account logon to the named Server and the login dialog does not appear for subsequent logins.

Step	Action	Result
		
3	Click OK.	Launches Control Builder. Go to Step 8.
4	The default User Name is usually the one used for the Windows account logon. For example, ps_user. The Station operator based default name is >mngr< and it is case sensitive. Please key in your assigned user name, as applicable.	Identifies user and associated security level.
5	Click and key in your login password in the Password field. You need an access level of at least Engineer to create a control strategy. Please check with your system administrator to get your assigned password, if required. The password is tied to your Windows account logon or the operator-based security for the Station application. The default password for operator-based security is >mngr1<, and it is case sensitive.	Confirms the identity of the user and associated security level.

Step	Action	Result
6	The Domain Name field identifies the domain where the Server is located. For Station operator-based security logon, the Domain Name must be Experion Security. Key in desired domain name or click the down-arrow button to select it from the list.	Identifies the Domain associated with the Server.
7	Click OK.	Launches Control Builder.
8	Go to the next section *Layout of the main window*.	Must open tree windows to view database and library contents.

Layout of the main window

The following figure shows the layout of Control Builder's main window.



To learn about the:	Click here
Menus	Click here
Toolbar	Click here
Control Drawing	Click here
Tree Window	Click here
Status Line	Click here

Menus

Control Builder's menus and their default submenus are listed in the table below. The actual list may vary depending on the nature of the activity you are currently attempting.

Menu	Description	
File	Allows access to the following submenus:	
	- Open - Close - Page Setup… - Export … - Exit	- New ->\ - Save - Print ->\ - Import…
Edit	Allows access to the following submenus:	
	- Copy - Delete… - Rename - Block Properties… - Execution Environment Assignment… - Links…	- Paste - Force Delete… - Module Properties… - Type - Module Containment…
View	Allows access to the following submenus:	

	- Toolbar - Project/Monitor Tree - Derivation View - Containment View - Type - License Display…	- Status Bar - Library Tree - Assignment View - Expand Item - Enable Tree Monitoring
Tools	The Tools menu allows access to a variety of submenu commands that are enabled/displayed based on the user's current activity. Some menu items that are exposed:	
	- Point Selection… - Resolve Substitute Connections… - Bulk Build - Validate Contents… - Convert PID to PID_PL… - User Preferences… - Checkpoint Preferences…	- Substitute Name List… - Bulk Edit Parameters -> - QVCS Manager… - Symbol Library… - System Preferences… - Operator Permissions…
Chart	Allows access to the following submenus:	

	- Configure Chart Size…	- Insert ->
	- View Grid	- View Page Breaks
	- Substitute Name List…	- Resolve Substitute Connections…
	- Re-Route Wire(s)	- Enable Auto Routing
	- Snap to Grid	- Show Delete Confirmation Connections
	- Delete SCM Handler	- Move Handler Left
	- Move Handler Right	- Set Invoke Transition
	- Cancel Chart Automatic Tracking	- Resume Chart Automatic Tracking
	- Disable Chart Monitoring	- Enable Chart Monitoring
Templates	Allows access to the following submenus:	
	- Make Template…	- Instantiate
	- Change Parent Template…	
Field Devices	Allows access to the following submenus:	
	- Device Match	- Device Unmatch
	- Device Replacement	- Methoids Manager…
Controller	Allows access to the following submenus:	

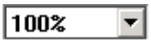
	- Activate - Change State… - Sequential Control Module Status… - Allow Database Changes - Load With Contents… - Delete Server Points… - Upload - Update to Project - Migrate…	- Inactivate -> - Checkpoint -> - Compare Parameters… - Load… - Load Server Points… - Upload Server Config. Data… - Upload With Contents… - Update With Contents (to Project)
Window	Allows access to the following submenus:	
	- Cascade - Arrange Icons	- Tile
Help	Allows access to the following submenus:	
	- Contents - Knowledge Builder - About Control Builder…	- Using Help - Knowledge Builder Search

Toolbar

The toolbar provides quick access to commonly used commands.

Button	Description
	Open Tree. Open a new tree window in Control Builder.
	Close. Closes the tree window that has focus in Control Builder.
	Open Library
	Left
	Right
	Save. Saves the item.
	Delete. Delestes the item that has focus.
	Copy. Copies the item to the clipboard.
	Paste. Pastes the item from the clipboard.
	Print. Prints the item that has focus.

Button	Description
	About
	Help. Displays the Help for Control Builder.
	Point Selection
	Wire
	Param Connector
	Execution Environment Assignment
	Module Containment
	Load
	UpLoad
	Substitute Names. Activates the dialog box that allows the assigning of substitute names for blocks.
	QVCS Manager. Activates QVCS Manager.

Button	Description
	Toggle State. Toggles the selected item to the opposite state, such as: inactive to active or acvtive to inactive.
	SCM Navigation
	Fieldbus Device Description Input
	Change Scale. Changes the scale of the item that has focus.

Control Drawing

The main work area of Control Builder where:

- Control Modules are created
- Function blocks are inserted and connected
- Control strategies are initiated

Status Line

The Status Line indicates a number of status properties of the Control Builder application, or a prompt for action.



At the left side of the status bar are prompts to help you to access features in Control Builder. At the right are four boxes that show various properties on status and connections to servers and their databases, which are described in the following table.

Status Line box (from left to right)	Description	
ERDB Synchronization status	Indicates the synchronization status of Control Builder with the ERDB. Valid values are:	
	Indication	**Color**
	<blank>	Grey
	SYNC	Green
	NOSYNC	Reverse video
	SYNCERR	Red
Connection to Server ERDB	Indicates whether Control Builder is connected to the **PRIMARY** (ServerB) or **BACKUP** (ServerA) ERDB. NOTE: You will not be able to build control strategies and perform most downloads when connected to the BACKUP server.	
Server	Shows the Server to which Control Builder is 'logged in.' (Derived from Configuration	

Status Line box (from left to right)	Description
	Studio.) By default, it also indicates the CDA Server which Control Builder is using.
Security Level Access	Indicates the user level access to which Control Builder is 'logged in.' For example, **ENGR** or **MNGR**.

Opening and navigating a tree window

This procedure illustrates how to initialize the tree windows within Control Builder.

Prerequisites:

- Control Builder is running

Considerations:

If this is the first time Control Builder has been launched, there are no tree windows open. Otherwise, Control Builder opens with the same tree windows that were displayed when it was last closed.

There are three available trees or views within each tree window:

- Project tree
- Monitoring tree
- Library tree

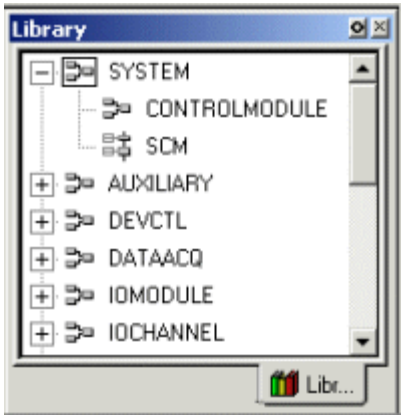


ATTENTION

Only **three** tree windows may be opened at any one time.

Step	Action	Result
1	Click View -> Project/Monitor Tree, OR Library Tree OR click the  Open Tree button in the toolbar.	Project/Monitor tree OR Library Tree window opens on the main Control Builder window.

Step	Action	Result
	(This option will not be available, if two tree windows are already open.)	
2	Repeat Step 1 to open another tree window, if required. (This option will not be available, if two tree windows are already open.)	A second tree window opens, as shown in the following figure.
		
3	Click the appropriate tab (Project, Monitoring or Library) on the bottom of the tree window to select the desired tree view.	Selected tree is displayed.

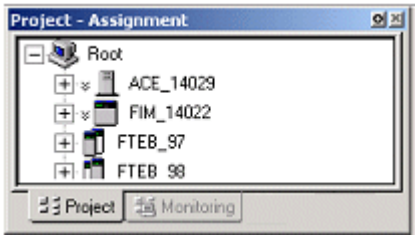
Step	Action	Result
4	Click on the ⊞ sign of the desired Library to expand and view the contents.	

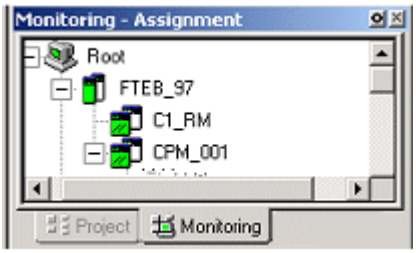
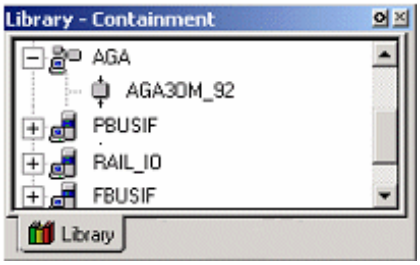
TIP

To expand the width of a tree window, move the cursor over its border until the cursor becomes a two-headed arrow, then drag to expand the window.

Tree Window tabs

The Tree Window provides quick access to three tabs: Project, Monitoring, and Library.

Tab	Description
	Project tab. The Project Tree View allows instances or strategies to be viewed as trees, showing the assignment relationships in a window in the Control Builder.

Tab	Description
	Monitoring tab. The Monitoring Tree View allows instances or strategies to be viewed as trees, showing the assignment relationships in a window in the Control Builder.
	Library tab The Library Tree View includes a variety of specific libraries that contains unique functions specific to that library.

Changing the state of a tree window

Prerequisites:

- Control Builder is running
- At least one tree window is open

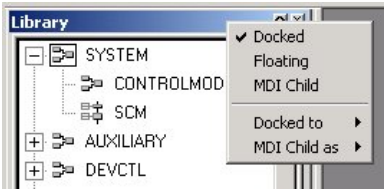
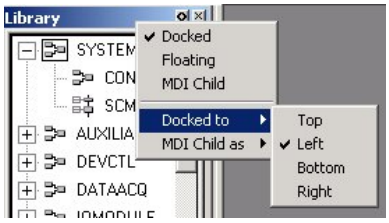
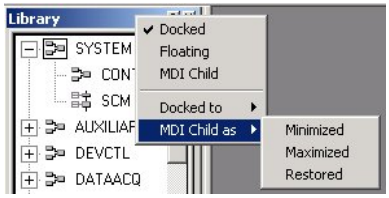
Considerations:

Tree windows can exist in three different states:

- ***Docked -*** the tree window is attached to one of the edges of the main Control Builder window. Tree windows in this state will not participate in normal child window cascades, sizing, or tiling operations.
- ***Floating -*** the tree window floats on top of or completely outside of the main Control Builder window. Tree windows in this state will not participate in normal child window cascade, sizing, or tiling operations.

- ***MDI Child -*** the tree window acts exactly like an open chart window. Tree windows in this state can be maximized, minimized, tiled and cascaded along with other Control Builder child windows (charts).

Perform the following procedure to change the state of a tree window.

Step	Action	Result
1	Right-click in the title bar area of the tree window. Alternate method: Hold the left-mouse button down with the cursor on top of the "diamond" button in the title bar.	A context menu pops up which enables you to change the state of the tree window. 
2	Select the desired state for the tree window.	Tree window assumes the desired state.
2A	For a docked tree window, select the sub-menu titled **Docked to** on the Context Menu to select a desired docking position (top, left, bottom, right).	Tree window assumes the desired docking position. 
2B	For an MDI Child tree window, select the sub-menu titled **MDI Child as** on the Context Menu to select a desired window size (minimized, maximized, restored).	Tree window assumes the desired MDI Child window size. 

TIP

The following shortcuts can also be used for changing the state of a tree window:

- Double-click on the tree window title bar to change a docked window to a floating window and vice versa.
- Click on the “diamond” button in the tree window title bar to toggle between a docked window and a regular MDI child window.
- Drag a docked or floating tree window near the edge of the main Control Builder window to dock the window in the selected position.
- Hold down the <Ctrl> key while dragging a docked or floating tree window to force the window to float.

Moving a docked tree window

Move a docked tree window using the following procedure:

Step	Action	Result
1	Click and drag the title bar of the appropriate tree window to another border of the main Control Builder window.	Tree window assumes the desired position. For example, if you drag the tree window near the right edge, it will “dock” on the right-hand side.

Closing a tree window

Close a tree window using the following procedure:

Step	Action	Result
1	Select either tree window by clicking on one of the tree window tabs or the title bar.	Window is highlighted.
2	Click File -> Close, or you can click the  Close tree button in the toolbar.	The selected tree window closes.

Block and configuration form tabs tables

The following section lists the available:

- blocks from the Library tab
- the tabs available on the configuration form for each block, and
- a link to an example of each tab type



ATTENTION

The following configuration form information is primarily based on PMIO.

Although Series C I/O information is included in the following, for more thorough Series C I/O configuration information, refer to the Series C I/O User's Guide

SYSTEM

Available library blocks and tabs

The following table lists:

- The individual block names in a two-columned list. Each column denotes the block's tabs on their Configuration form.
- The tabs available for each block's Configuration form.
- A link to an example of each tab.
  Note: Main tabs typically, are unique for each block, therefore, a link to an example of a Main tab is provided. There may be a slight variation between the example and the actual tab.
- N/A = tab is not available for block's configuration form
- No example = no example of tab in document

Tab name	Block names - two different tab sets	
	System	**SCM**
Main	Click here	Click here
Arbitration	N/A	No example
Handlers	N/A	No example
Alarm and Event	N/A	Click here
Recipe	N/A	No example
History	N/A	No example
Aliases	N/A	No example
Server History	Click here	Click here
Server Displays	Click here	Click here
Control Confirmation	Click here	Click here
Status	N/A	No example
QVCS	Click here	Click here
Identification	Click here	Click here
Dependencies	Click here	N/A
Projected Parameters	Click here	Click here
Block Pins	Click here	Click here
Configuration Parameters	Click here	Click here
Monitoring Parameters	Click here	Click here
Block Preferences	Click here	Click here
Template Defining	Click here	N/A

AUXILIARY

Available library blocks and tabs

The following table lists:

- The individual block names that contain the same named tabs on their Configuration form.
- The tabs available for each block's Configuration form.
- A link to an example of each tab.
  Note: Main tabs typically, are unique for each block, therefore, a link to an example of a Main tab is provided. There may be a slight variation between the example and the actual tab.

Tab name	Block names - all having the same-named tabs
	• **DEADTIME** • **GENLIN** • **LEADLAG** • **TOTALIZER**
Main	Click here
Identification	Click here
Dependencies	Click here
Block Pins	Click here
Configuration Parameters	Click here
Monitoring Parameters	Click here
Block Preferences	Click here
Template Defining	Click here

DEVCTL

Available library blocks and tabs

The following table lists:

- The individual block name
- The tabs available for the Configuration form
- A link to an example of each tab.
  Note: Main tabs typically, are unique for each block, therefore, a link to an example of a Main tab is provided. There may be a slight variation between the example and the actual tab.

Tab name	Block name
	DEVCTL
Main	Click here
Identification	Click here
Dependencies	Click here
Block Pins	Click here
Configuration Parameters	Click here
Monitoring Parameters	Click here
Block Preferences	Click here
Template Defining	Click here

DATAACQ

Available library blocks and tabs

The following table lists:

- The individual block name
- The tabs available for the block's Configuration form
- A link to an example of each tab.
  Note: Main tabs typically, are unique for each block, therefore, a link to an example of a Main tab is provided. There may be a slight variation between the example and the actual tab.

Tab name	Block name
	DATAACQ
Main	Click here
Alarms	Click here
Identification	Click here
Dependencies	Click here
Block Pins	Click here
Configuration Parameters	Click here
Monitoring Parameters	Click here
Block Preferences	Click here
Template Defining	Click here

HIWAY

Available library blocks and tabs

The following table lists:

- The individual block names
- The tabs available for each block's Configuration form
- A link to an example of each tab.
  Note: Main tabs typically, are unique for each block, therefore, a link to an example of a Main tab is provided. There may be a slight variation between the example and the actual tab.

Tab name	Block names - four different tab sets			
	AMC	**AMCRCD**	**CB**	**CBRCD**
Main	Click here	Click here	Click here	Click here
Device Status	No example	No example	No example	No example
Slot 1-8	No example	N/A	N/A	N/A
Slot 9-16	No example	N/A	N/A	N/A
Regulatory FBs	No example	N/A	N/A	N/A
UAC Information	N/A	No example		N/A
Slot Configuration	N/A	N/A	No example	N/A
Slot Tog Intervals	No example	N/A	N/A	N/A
Slot Status Data	No example	N/A	N/A	N/A
UAC Information 1	N/A	N/A	N/A	No example
UAC Information 2	N/A	N/A	N/A	No example
QVCS	Click here	Click here	Click here	Click here
Server History	Click here	Click here	Click	Click here

			here	
Server Displays	Click here	Click here	Click here	Click here
Control Confirmation	Click here	Click here	Click here	Click here
Identification	Click here	Click here	Click here	Click here

Tab name	Block names - four different tab sets		
	DHP	HLPIU	LEPIU LLPIU
Main	Click here	Click here	Click here
Device Status	No example	No example	No example
PC Configuration 1	No example	N/A	N/A
PC Configuration 2	No example	N/A	N/A
Slot 1-16	No example	N/A	No example
Slot 17-32	No example	N/A	No example
Sub Slot Configuration	No example	N/A	No example
Slot 1-8	N/A	No example	N/A
Slot 9-16	N/A	No example	N/A
Slot 17-24	N/A	No example	N/A
Slot 25-32	N/A	No example	N/A
QVCS	Click here	Click here	Click here
Server History	Click here	Click here	Click here
Server Displays	Click here	Click here	Click here
Control Confirmation	Click here	Click here	Click here
Identification	Click here	Click here	Click here

HIWAYIF

Available library blocks and tabs

The following table lists:

- The individual block names
- The tabs available for each block's Configuration form
- A link to an example of each tab.
  Note: Main tabs typically, are unique for each block, therefore, a link to an example of a Main tab is provided. There may be a slight variation between the example and the actual tab.

Tab name	Block names - all having the same-named tabs	
	• **EHGOUT**	• **HIWAYOUT**
Main	Click here	
Module Configuration	Click here	
Channel Configuration	Click here	
Status/Data	Click here	
QVCS	Click here	
Server History	Click here	
Server Displays	Click here	
Control Confirmation	Click here	
Identification	Click here	

IOMODULE

Available library blocks

The following table lists:

- The individual block names that contain the same named tabs on their Configuration form.
- The tabs available for each block's Configuration form.
- A link to an example of each tab.
  Note: Main tabs typically, are unique for each block, therefore, a link to an example of a Main tab is provided. There may be a slight variation between the example and the actual tab.

Tab name	Block names - all having the same named tabs		
	• **TC-IAH061** • **TC-IAH161** • **TC-IDA161** • **TC-IDD321** • **TC-IDJ161** • **TC-IDK161**	• **TC-IDW161** • **TC-IXL061** • **TC-IXR061** • **TC-MUX021** • **TC-OAH061** • **TC-OAV061**	• **TC-ODA161** • **TC-ODD321** • **TC-ODJ161** • **TC-ODK161** • **TC-ORC081** • **TC-ORC161**
Main	Click here		
Module Configuration	Click here		
Channel Configuration	Click here		
Version	Click here		
Server History	Click here		
Server Displays	Click here		
Control Confirmation	Click here		
Identification	Click here		

The following table lists:

- The individual block names in a two-columned list. Each column denotes blocks having similar tabs on their Configuration form.
- N/A = tab is not available for block's configuration form

Tab name	Block names - two different tab sets	
	• **TC-IDX081** • **TC-IDX161** • **TC-ODX081** • **TC-ODX161**	**TC-OAV081**
Main	Click here	Click here
Module Configuration	Click here	Click here
Channel Configuration	Click here	Click here
Diagnostic Configuration	Click here	Click here
Status/Data	N/A	Click here
Version	Click here	Click here
Server History	Click here	Click here
Server Displays	Click here	Click here
Control Confirmation	Click here	Click here
Identification	Click here	Click here

IOCHANNEL

Available library blocks and tabs

The following table lists:

- The individual block names in a two-columned list. Each column denotes blocks having similar tabs on their Configuration forms.
- The tabs available for each block's Configuration form
- A link to an example of each tab.
  Note: Main tabs typically, are unique for each block, therefore, a link to an example of a Main tab is provided. There may be a slight variation between the example and the actual tab.
- N/A = tab is not available for block's configuration form
- No example = no example of tab in document

Tab name	Block names - two different tab sets	
	• **AICHANNEL** • **AOCHANNEL** • **DICHANNEL** • **DOCHANNEL** • **PWMCHANNEL**	• **SIFLAGARRCH** • **SINUMARRCH** • **SITEXTARRCH**
Main	Click here	Click here
Serial Device	N/A	No example
Status/Data	N/A	Click here
Identification	Click here	Click here
Dependencies	Click here	Click here
Block Pins	Click here	Click here
Configuration Parameters	Click here	Click here
Monitoring	Click here	Click here

Parameters		
Block Preferences	Click here	Click here
Template Defining	Click here	Click here

LIOM_FB

Available library blocks and tabs

The following table lists:

- The individual block names
- The tabs available for each block's Configuration form
- A link to an example of each tab.
  Note: Main tabs typically, are unique for each block, therefore, a link to an example of a Main tab is provided. There may be a slight variation between the example and the actual tab.

Tab name	Block names - all having the same-named tabs	
	• **ABS** • **ADD** • **AND** • **AUXCALC** • **BITAND** • **BITOR** • **BITRD** • **BITWR** • **BITXOR** • **DATAACQ** • **DELAY** • **DEVCTL** • **DIV** • **EQ**	• **LCSFLAG** • **LCSNUMERIC** • **LCSNUMERIC16** • **LCSOFFDELAY** • **LCSONDELAY** • **LCSRS** • **LCSSEL** • **LCSTYPECONVERT** • **LE** • **LT** • **MUL** • **NE** • **NOT** • **NUMERICARRAY**

	• **FLAGARRAY** • **FTRIG** • **GE** • **GT** • **LCS1SCOMPLEMENT** • **LCSCOUNTER**	• **OR** • **PUSH** • **RTRIG** • **SQRT** • **SUB** • **XOR**
Main	Click here	
Module Configuration	Click here	
Channel Configuration	Click here	
Status/Data	Click here	
QVCS	Click here	
Server History	Click here	
Server Displays	Click here	
Control Confirmation	Click here	
Identification	Click here	

LIOM_IOC

Available library blocks and tabs

The following table lists:

- The individual block names
- The tabs available for each block's Configuration form
- A link to an example of each tab.
  Note: Main tabs typically, are unique for each block, therefore, a link to an example of a Main tab is provided. There may be a slight variation between the example and the actual tab.

Tab name	Block names - all having the same-named tabs	
	• **LCSAEMAOCHANNEL** • **LCSAIMAOCHANNEL** • **LCSAOMAOCHANNEL**	• **LCSDIMAOCHANNEL** • **LCSDOMAOCHANNEL**
Main	Click here	
Module Configuration	Click here	
Channel Configuration	Click here	
Status/Data	Click here	
QVCS	Click here	
Server History	Click here	
Server Displays	Click here	
Control Confirmation	Click here	
Identification	Click here	

LIOM_IOM

Available library blocks and tabs

The following table lists:

- The individual block names
- The tabs available for each block's Configuration form
- A link to an example of each tab.
  Note: Main tabs typically, are unique for each block, therefore, a link to an example of a Main tab is provided. There may be a slight variation between the example and the actual tab.

Tab name	Block names - four different tab sets			
	621-0007RC **621-1101-RC** **621-0010-VRC** **621-1160RC** **621-1250RC** **621-2101RC** **621-2102RC** **621-2150RC** **621-2175RC** **621-2200RC** **621-3300RC** **621-3560RC** **621-3580RC** **621-4350RC** **621-6503RC** **621-6550RC**	**621-0010-ARC** **621-0010-VRC** **621-0018R** **621-0022AR** **621-0022VR**	**621-0020RC** **621-0025RC**	**621-0021RC**

	Block names - four different tab sets			
	621-6575RC			
Main	Click here	Click here	Click here	Click here
Module Configuration	Click here	Click here	Click here	Click here
Channel Configuration	Click here	N/A	Click here	N/A
Data	N/A	Click here	N/A	N/A
Channel Status	N/A	N/A	No example	N/A
QVCS	Click here	Click here	Click here	Click here
Server History	Click here	Click here	Click here	Click here
Server Displays	Click here	Click here	Click here	Click here
Control Confirmation	Click here	Click here	Click here	Click here
Identification	Click here	Click here	Click here	Click here

PMIO

Available library blocks and tabs

The following table lists:

- The individual block names in a columned list. Each column denotes blocks having similar tabs on their Configuration form.
- The tabs available for each block's Configuration form
- A link to an example of each tab.
  Note: Main tabs typically, are unique for each block, therefore, a link to an example of a Main tab is provided. There may be a slight variation between the example and the actual tab.
- N/A = tab is not available for block's configuration form
- No example = no example of tab in document

Tab name	Block names - two different tab sets	
	• **AO16 AOCHANNEL** • **AO8 AOCHANNEL** • **DIDI CHANNEL** • **DI24V DICHANNEL** • **DISOE DICHANNEL** • **DO16 DOCHANNEL** • **DO32 DOCHANNEL**	• **HLAI AICHANNEL** • **LLAIAI CHANNEL** • **LLMUX AICHANNEL** • **RHMUX AICHANNEL** • **STIMV AICHANNEL**
Main	Click here	Click here
Configuration	No example	No example
Smart Transmitter	N/A	Click here

Identification	Click here	Click here
Dependencies	Click here	Click here
Block Pins	Click here	Click here
Configuration Parameters	Click here	Click here
Monitoring Parameters	Click here	Click here
Block Preferences	Click here	Click here
Template Defining	Click here	Click here

Tab name	Block names - all having the same named tabs
	• **AO16HART HAOCHANNEL** • **HLAIHART HAICHANNEL**
Main	Click here
Configuration	No example
HART Confirmation	No example
HART Device Status	No example
HART Identification	No example
HART Variables	No example
HART Notifications	No example
Configuration Parameters	Click here
Monitoring Parameters	Click here
Block Preferences	Click here
Template Defining	Click here

Version	Click here
Server History	Click here
Server Displays	Click here
Control Confirmation	Click here
Identification	Click here
Dependencies	Click here
Block Pins	Click here

LOGIC

Available library blocks and tabs

The following table lists:

- The individual block names that contain the same named tabs on their Configuration form.
- The tabs available for each block's Configuration form.
- A link to an example of each tab.
  Note: Main tabs typically, are unique for each block, therefore, a link to an example of a Main tab is provided. There may be a slight variation between the example and the actual tab.

Tab name	Block names - all having the same named tabs		
	• **2003** • **AND** • **CHECKBAD** • **DELAY** • **EQ** • **FTRIG** • **GE** • **GT** • **LE** • **LIMIT** • **LT** • **MAX** • **MAXPULSE** • **MIN**	• **MINPULSE** • **MUX** • **MUXREAL** • **MVOTE** • **NANAD** • **NE** • **NOON** • **NOR** • **NOT** • **OFFDELAY** • **ONDELAY** • **OR** • **PULSE** • **QOR**	• **ROL** • **ROR** • **RS** • **RTRIG** • **SEL** • **SELREAL** • **SHL** • **SHR** • **SR** • **TRIG** • **WATCHDOG** • **XOR**
Main	Click here		
Identification	Click here		

	Block names - all having the same named tabs
Dependencies	Click here
Block Pins	Click here
Configuration Parameters	Click here
Monitoring Parameters	Click here
Block Preferences	Click here
Template Defining	Click here

REGCTL

Available library blocks and tabs

The following table lists:

- The individual block names that contain the same named tabs on their Configuration form.
- The same tabs are available for each block's Configuration form.
- A link to an example of each tab.
  Note: Main tabs typically, are unique for each block, therefore, a link to an example of a Main tab is provided. There may be a slight variation between the example and the actual tab.
- No example = no example of tab in document

Tab name	Block names - all having the same named tabs
	• **PULSECOUNT** • **PULSELENGTH** • **REEOUT**
Main	Click here
Identification	Click here
Dependencies	Click here
Block Pins	Click here
Configuration Parameters	Click here
Monitoring Parameters	Click here
Block Preferences	Click here
Template Defining	Click here

Tab name	Block names - all having the same named tabs
	• **AUTOMAN**

	• **RATIOBAIAS** • **REMCAS**
Main	Click here
Output	No example
Alarms	Click here
SCM	No example
Identification	Click here
Dependencies	Click here
Block Pins	Click here
Configuration Parameters	Click here
Monitoring Parameters	Click here
Block Preferences	Click here
Template Defining	Click here

Available library blocks and tabs

The following table lists:

- The individual block names in a columned list. Each column denotes blocks having similar tabs on their Configuration form.
- The tabs available for each block's Configuration form
- A link to an example of each tab.
  Note: Main tabs typically, are unique for each block, therefore, a link to an example of a Main tab is provided. There may be a slight variation between the example and the actual tab.
- N/A = tab is not available for block's configuration form
- No example = no example of tab in document

Tab name	Block names - three different tab sets		
	FANOUT	• **OVRDSEL** • **SWITCH**	**POSPROP**
Main	Click here	Click here	Click here
Common Output	No example	N/A	N/A
Individual Output	No example	N/A	N/A
Input	N/A	No example	N/A
Output	N/A	No example	N/A
Algorithm	N/A	N/A	No example
SetPoint	N/A	N/A	No example
Alarms	Click here	Click here	Click here
SCM	No example	No example	No example
Identification	Click here	Click here	Click here
Dependencies	Click here	Click here	Click here
Block Pins	Click here	Click here	Click here

Configuration Parameters	Click here	Click here	Click here
Monitoring Parameters	Click here	Click here	Click here
Block Preferences	Click here	Click here	Click here
Template Defining	Click here	Click here	Click here

Tab name	**Block names - four different tab sets**			
	PID	**PIDFF**	**PID-PL**	**RAMPSOAK**
Main	Click here	Click here	Click here	Click here
Algorithm	No example	No example	No example	N/A
SetPoint	No example	No example	No example	N/A
Profile	N/A	N/A	N/A	No example
Profile Graph	N/A	N/A	N/A	No example
Active Profile Graph	N/A	N/A	N/A	No example
Output	No example	No example	No example	No example
Advanced	N/A	N/A	No example	N/A
Alarms	Click here	Click here	Click here	Click here
SCM	No example	No example	No example	No example
Identification	Click here	Click here	Click here	Click here
Dependencies	Click here	Click here	Click here	Click here
Block Pins	Click here	Click here	Click here	Click here
Configuration Parameters	Click here	Click here	Click here	Click here
Monitoring	Click here	Click here	Click here	Click here

Parameters				
Block Preferences	Click here	Click here	Click here	Click here
Template Defining	Click here	Click here	Click here	Click here

SCM

Available library blocks

The following table lists:

- The individual block names in a three-columned list. Each column denotes blocks having similar tabs on their Configuration form.
- The tabs available for each block's Configuration form
- A link to an example of each tab.
  Note: Main tabs typically, are unique for each block, therefore, a link to an example of a Main tab is provided. There may be a slight variation between the example and the actual tab.
- N/A = tab is not available for block's configuration form
- No example = no example of tab in document

	Block names - three different tab sets		
Tab name	**HANDELER**	**STEP**	**TRANSITION**
Main	Click here	Click here	Click here
Output #1	N/A	No example	N/A
Stop Condition	N/A	No example	N/A
Interrupt Condition	N/A	No example	N/A
Gates	N/A	N/A	No example
Identification	Click here	Click here	Click here
Block Pins	N/A	Click here	Click here
Configuration Parameters	N/A	Click here	Click here

Monitoring Parameters	N/A	Click here	Click here
Block Preferences	N/A	Click here	Click here

SERIES_C_IO

Available library blocks and tabs

The following table lists:

- The individual block names
- The tabs available for each block's Configuration form
- A link to an example of each tab.
  Note: Main tabs typically, are unique for each block, therefore, a link to an example of a Main tab is provided. There may be a slight variation between the example and the actual tab.

Tab name	Block names - all having the same-named tabs		
	• **AI-HART** • **AO-HART**	• **AI-LLMUX**	• **DI-24** • **DI-HV** • **DO-24B**
Main	Click here	Click here	Click here
HART Status	No example	N/A	N/A
Status/Data	Click here	Click here	Click here
Maintenance	No example	No example	No example
Calibration	No example	No example	N/A
Box Soft Failures	No example	No example	No example
Channel Soft Failures	No example	No example	No example
QVCS	Click here	Click here	Click here
Server History	Click here	Click here	Click here
Server Displays	Click here	Click here	Click here

Control Confirmation	Click here	Click here	Click here
Identification	Click here	Click here	Click here

UCNIF

Available library block and tabs

The following table lists:

- The individual block name
- The tabs available for each block's Configuration form
- A link to an example of each tab.
  Note: Main tabs typically, are unique for each block, therefore, a link to an example of a Main tab is provided. There may be a slight variation between the example and the actual tab.

Tab name	Block name
	UCNOUT
Main	Click here
Identification	Click here
Dependencies	Click here
Block Pins	Click here
Configuration Parameters	Click here
Monitoring Parameters	Click here
Block Preferences	Click here
Template Defining	Click here

UTILITY

Available library blocks and tabs

The following table lists:

- The individual block names that contain the same named tabs on their Configuration form.
- The tabs available for each block's Configuration form.
- A link to an example of each tab.
  Note: Main tabs typically, are unique for each block, therefore, a link to an example of a Main tab is provided. There may be a slight variation between the example and the actual tab.

Tab name	Block names - all having the same named tabs		
	• **FLAG** • **FLAGARRAY** • **MESSAGE**	• **NUMERIC** • **NUMERICARAY** • **PUSH**	• **TEXTARRAY** • **TIMER** • **TYPECONVERT**
Main	Click here		
Identification	Click here		
Dependencies	Click here		
Block Pins	Click here		
Configuration Parameters	Click here		
Monitoring Parameters	Click here		
Block Preferences	Click here		
Template Defining	Click here		

FIELDBUS

Available library blocks and tabs

The following table lists:

- The individual block names that contain the same named tabs on their Configuration form.
- The tabs available for each block's Configuration form.
- A link to an example of each tab.
  Note: Main tabs typically, are unique for each block, therefore, a link to an example of a Main tab is provided. There may be a slight variation between the example and the actual tab.
- No example = no example of tab in document

Tab name	Block names - all having the same named tabs		
	• **AI** • **ANALOG_ ALARM** • **AO** • **ARITHMETIC** • **DI**	• **DO** • **INPUT_ SELECTOR** • **INTEGRATOR** • **LEADLAG** • **OUTPUT_ SPLITTER**	• **PID** • **SETPOINT_ RAMP_ GENERATION** • **SIGNAL_ CHARACTERIZER** • **TIMER**
Process	No example		
Alarm	Click here		
Maintenance	No example		
Tune	Click here		
Ranges	No example		
Other	No example		
Identification	Click here		
Dependencies	Click here		

Block Pins	Click here
Configuration Parameters	Click here
Monitoring Parameters	Click here
Block Preferences	Click here
Template Defining	Click here

HARTIO

Available library blocks and tabs

The following table lists:

- The individual block names in a three-columned list. Each column denotes blocks having similar tabs on their Configuration form.
- The tabs available for each block's Configuration form
- A link to an example of each tab.
  Note: Main tabs typically, are unique for each block, therefore, a link to an example of a Main tab is provided. There may be a slight variation between the example and the actual tab.
- N/A = tab is not available for block's configuration form
- No example = no example of tab in document

Tab name	**Block names - three different tab sets**		
	• **HARTDEVICE** • **LOGIX1200_0104** • **ST3000_0103** • **ST3000_0202** • **STT25H_0103**	**TC-HAI081**	**TC-HAO081**
Main	Click here	Click here	Click here
Device Information	No example	N/A	N/A

Dynamic Variables	No example	N/A	N/A
Device Variables	No example	N/A	N/A
Module Configuration	N/A	Click here	Click here
Channel Configuration	N/A	Click here	Click here
Device Information	N/A	No example	No example
HART Configuration	N/A	No example	No example
Status/Data	N/A	N/A	Click here
Version	Click here	Click here	Click here
Server History	Click here	Click here	Click here
Server Displays	Click here	Click here	Click here
Control Confirmation	Click here	Click here	Click here
Identification	Click here	Click here	Click here

DNETIF

Available library blocks and tabs

The following table lists:

- The individual block names in a columned list. Each column denotes the block's tabs on their Configuration form.
- The tabs available for each block's Configuration form
- A link to an example of each tab.
  Note: Main tabs typically, are unique for each block, therefore, a link to an example of a Main tab is provided. There may be a slight variation between the example and the actual tab.
- N/A = tab is not available for block's configuration form
- No example = no example of tab in document

Tab name	Block names - two different tab sets	
	DNET_DEVICE	DNET_IM
Main	Click here	Click here
Module Configuration	Click here	Click here
Interface Module	N/A	No example
Device Configuration/Status	N/A	No example
Command/Status	N/A	No example
Version	Click here	Click here
Server History.	Click here	Click here
Server Displays	Click here	Click here
Control Confirmation	Click here	Click here
Identification	Click here	Click here

Tab name	Block names - two different tab sets	
	DNET_INCHAN	DNET_OUTCHAN
Main	Click here	Click here
Input Data Location	No example	N/A
Numeric Input Config.	No example	N/A
Discrete Input Config.	No example	N/A
Output Data Location	N/A	No example
Numeric Output Config.	N/A	No example
Discrete Output Config.	N/A	No example

Numeric Data	No example	No example
Discrete Data	No example	No example
Identification	Click here	Click here
Dependencies	Click here	Click here
Block Pins	Click here	Click here
Configuration Parameters	Click here	Click here
Monitoring Parameters	Click here	Click here
Block Preferences	Click here	Click here
Template Defining	Click here	Click here

AB_DRIVE_IF

Available library blocks and tabs

The following table lists:

- The individual block names that contain the same named tabs on their Configuration form.
- The tabs available for each block's Configuration form
- A link to an example of each tab.
  Note: Main tabs typically, are unique for each block, therefore, a link to an example of a Main tab is provided. There may be a slight variation between the example and the actual tab.
- No example = no example of tab in document

Tab name	Block names - all having the same named tabs
	• **1305** • **1336-PLUS-II** • **GENERIC_DRIVE**
Main	Click here

Module Configuration	Click here
Status/Data	Click here
Datalink Parameters	No example
Version	Click here
Server History.	Click here
Server Displays	Click here
Control Confirmation	Click here
Identification	Click here

The following table lists:

- The individual block names in a two-columned list. Each column denotes the block's tabs on their Configuration form.
- The tabs available for each block's Configuration form
- A link to an example of each tab.
  Note: Main tabs typically, are unique for each block, therefore, a link to an example of a Main tab is provided. There may be a slight variation between the example and the actual tab.
- N/A = tab is not available for block's configuration form
- No example = no example of tab in document

Tab name	**Block names - two different tab sets**	
	DRIVE_INPUT	**DRIVE_OUTPUT**
Main	Click here	Click here
Datalink Configuration	No example	No example
Drive Status	No example	N/A
Logic Control	N/A	No example
Datalink Status	No example	No example

Tab name	Block names - two different tab sets	
	DRIVE_INPUT	**DRIVE_OUTPUT**
Identification	Click here	Click here
Dependencies	Click here	Click here
Block Pins	Click here	Click here
Configuration Parameters	Click here	Click here
Monitoring Parameters	Click here	Click here
Block Preferences	Click here	Click here
Template Defining	Click here	Click here

AGA

Available library blocks and tabs

The following table lists:

- The individual block names
- The tabs available for each block's Configuration form
- A link to an example of each tab. There may be a slight variation between the example and the actual tab.
- N/A = tab is not available for block's configuration form
- No example = no example of tab in document

Tab name	Block names - five different tab sets				
	AGA 3OM_92	**AGA 7TM_96**	**AGA 8DL_94**	**AGA 8GS_94**	**AGA 9UM_98**
AGA 3 Orfice Meter	No example	N/A	N/A	N/A	N/A
Orifice Meter Setup	No example	N/A	N/A	N/A	N/A

Tab name	Block names - five different tab sets				
	AGA 3OM_92	**AGA 7TM_96**	**AGA 8DL_94**	**AGA 8GS_94**	**AGA 9UM_98**
AGA 7 Turbine Meter	N/A	No example	N/A	N/A	N/A
AGA 8 Detail Setup Data	N/A	N/A	No example	N/A	N/A
Gas Component Data	N/A	N/A	No example	N/A	N/A
AGA 8 Gross Setup Data	N/A	N/A	N/A	No example	N/A
Method Setup Data	N/A	N/A	N/A	No example	N/A
AGA 9 Ultrsonic Meter	N/A	N/A	N/A	N/A	No example
Identification	Click here	Click here	Click here	Click here	Click here
Dependencies	Click here	Click here	Click here	Click here	Click here
Block Pins	Click here	Click here	Click here	Click here	Click here
Configuration Parameters	Click here	Click here	Click here	Click here	Click here
Monitoring Parameters	Click here	Click here	Click here	Click here	Click here
Block Preferences	Click here	Click here	Click here	Click here	Click here
Template Defining	Click here	Click here	Click here	Click here	Click here

PBUSIF

Available library blocks and tabs

The following table lists:

- The individual block names
- The tabs available for each block's Configuration form

- A link to an example of each tab.
  Note: Main tabs typically, are unique for each block, therefore, a link to an example of a Main tab is provided. There may be a slight variation between the example and the actual tab.

Tab name	Block names - all having the same named tabs
	• **BIZERBASTDEV** • **ENCODERDEV** • **PBI_DEVICE** • **PROFIDRIVEDEV** • **SIMOCODE3UF5DEV**
Main	Click here
Module Configuration	Click here
Version	Click here
Server History.	Click here
Server Displays	Click here
Control Confirmation	Click here
Identification	Click here

The following table lists:

- The individual block names in a columned list. Each column denotes the block ‘s tabs on their Configuration form.
- The tabs available for each block’s Configuration form
- A link to an example of each tab.
  Note: Main tabs typically, are unique for each block, therefore, a link to an example of a Main tab is provided. There may be a slight variation between the example and the actual tab.
- N/A = tab is not available for block’s configuration form
- No example = no example of tab in document

Tab name	Block names - three different tab sets		
	BIZERBASTIN	**BIZERBASTOUT**	**ENCODERIN**
Main	Click here	Click here	Click here
Channel Status	No example	No example	N/A
Command Data	N/A	No example	N/A
Status/Data	N/A	N/A	Click here
Input/Process Data	No example	N/A	N/A
ST Status	No example	N/A	N/A
Function Status	No example	N/A	N/A
Status I/O	No example	N/A	N/A
Identification	Click here	Click here	Click here
Dependencies	Click here	Click here	Click here
Block Pins	Click here	Click here	Click here
Configuration Parameters	Click here	Click here	Click here
Monitoring Parameters	Click here	Click here	Click here
Block Preferences	Click here	Click here	Click here
Template Defining	Click here	Click here	Click here

Tab name	Block names - four different tab sets			
	ENCODEROUT	**PB DIAGNOSTIC**	**PBI_INCHAN**	**PBI_OUTCHAN**
Main	Click here	Click here	Click here	Click here
Diagnostic Data Location	N/A	No example	N/A	N/A
Input Data Location	N/A	N/A	No example	N/A

Tab name	Block names - four different tab sets			
	ENCODEROUT	**PB DIAGNOSTIC**	**PBI_INCHAN**	**PBI_OUTCHAN**
Numeric Input Config.	N/A	No example	No example	N/A
Discrete Input Config.	N/A	No example	No example	N/A
Output Data Location	N/A	N/A	N/A	No example
Numeric Output Config.	N/A	N/A	N/A	No example
Discrete Output Config.	N/A	N/A	N/A	No example
Numeric Data	N/A	No example	No example	No example
Discrete Data	N/A	No example	No example	No example
Commands/Data	No example	N/A	N/A	N/A
Identification	Click here	Click here	Click here	Click here
Dependencies	Click here	Click here	Click here	Click here
Block Pins	Click here	Click here	Click here	Click here
Configuration Parameters	Click here	Click here	Click here	Click here
Monitoring Parameters	Click here	Click here	Click here	Click here
Block Preferences	Click here	Click here	Click here	Click here
Template Defining	Click here	Click here	Click here	Click here

Tab name	Block names - two different tab sets	
	PROFIDRIVEIN	**PROFIDRIVEOUT**
Main	Click here	Click here
Input Configuration	No example	N/A

Tab name	Block names - two different tab sets	
	PROFIDRIVEIN	**PROFIDRIVEOUT**
Channel Status	No example	No example
Input/Process Data	No example	N/A
Status Word	No example	N/A
Output Configuration	N/A	No example
Parameter/Process Data	N/A	No example
Control Word	N/A	No example
Identification	Click here	Click here
Dependencies	Click here	Click here
Block Pins	Click here	Click here
Configuration Parameters	Click here	Click here
Monitoring Parameters	Click here	Click here
Block Preferences	Click here	Click here
Template Defining	Click here	Click here

Tab name	Block names - three different tab sets		
	PBIM_SST	• **SIMATIC_AI** • **SIMATIC_AO**	• **SIMATIC_DI** • **SIMATIC_DO**
Main	Click here	Click here	Click here
Module Configuration	Click here	Click here	Click here
PFB Station Config/Diag.	No example	N/A	N/A
PFB Module	No example	N/A	N/A

Tab name	Block names - three different tab sets		
	PBIM_SST	• **SIMATIC_AI** • **SIMATIC_AO**	• **SIMATIC_DI** • **SIMATIC_DO**
Config/Diag			
Commands/Status	No example	N/A	N/A
Statistics	No example	N/A	N/A
Channel Configuration	N/A	Click here	N/A
Status/Data	N/A	Click here	Click here
Version	Click here	Click here	Click here
Server History.	Click here	Click here	Click here
Server Displays	Click here	Click here	Click here
Control Confirmation	Click here	Click here	Click here
Identification	Click here	Click here	Click here

Tab name	Block names - two different tab sets	
	SIMOCODE3UF5IN	**SIMCODE3UF5OUT**
Main	Click here	Click here
Channel Config/Status	No example	N/A
Process Data	No example	N/A
Signal Data 0/1 Detail	No example	N/A
Signal Data 2/3 Detail	No example	N/A
Channel Status	N/A	No example
Control Data	N/A	No example
Control Data 0/1	N/A	No example

Tab name	Block names - two different tab sets	
	SIMOCODE3UF5IN	**SIMCODE3UF5OUT**
Detail		
Control Data 2/3 Detail	N/A	No example
Identification	Click here	Click here
Dependencies	Click here	Click here
Block Pins	Click here	Click here
Configuration Parameters	Click here	Click here
Monitoring Parameters	Click here	Click here
Block Preferences	Click here	Click here
Template Defining	Click here	Click here

RAILIO

Available library blocks and tabs

The following table lists:

- The individual block names
- The tabs available for each block's Configuration form
- A link to an example of each tab.
  Note: Main tabs typically, are unique for each block, therefore, a link to an example of a Main tab is provided. There may be a slight variation between the example and the actual tab.

Tab name	Block names - all having the same named tabs	
	• **TC-FIAH81** • **TC-FID161** • **TC-FIDA81** • **TC-FIL081** • **TC-FIR081**	• **TC-FOA041** • **TC-FOD161** • **TC-FODA81** • **TC-FOR081**
Main	Click here	
Module Configuration	Click here	
Channel Configuration	Click here	
Status/Data	Click here	
Version	Click here	
Server History	Click here	
Server Displays	Click here	
Control Confirmation	Click here	
Identification	Click here	

FBUSIF

Available library blocks and tabs

The following table lists:

- The individual block names in a columned list. Each column denotes the block's tabs on their Configuration form.
- The tabs available for each block's Configuration form
- A link to an example of each tab.
  Note: Main tabs typically, are unique for each block, therefore, a link to an example of a Main tab is provided. There may be a slight variation between the example and the actual tab.
- N/A = tab is not available for block's configuration form
- No example = no example of tab in document

Tab name	Block names - four different tab sets			
	IN	IN_D	OUT	OUT_D
Main	Click here	Click here	Click here	Click here
EU Range	N/A	N/A	No example	N/A
Identification	Click here	Click here	Click here	Click here
Dependencies	Click here	Click here	Click here	Click here
Block Pins	Click here	Click here	Click here	Click here
Configuration Parameters	Click here	Click here	Click here	Click here
Monitoring Parameters	Click here	Click here	Click here	Click here
Block Preferences	Click here	Click here	Click here	Click here
Template Defining	Click here	Click here	Click here	Click here

Tab name	Block names - all having the same named tabs

	1757-CN2FF
Main	Click here
Module Configuration	Click here
Channel Configuration	Click here
Status/Data	Click here
Version	Click here
Server History	Click here
Server Displays	Click here
Control Confirmation	Click here
Identification	Click here

EXCHANGE

Available library blocks and tabs

The following table lists:

- The individual block names in a columned list. Each column denotes blocks having similar tabs on their Configuration form.
- The tabs available for each block's Configuration form
- A link to an example of each tab.
  Note: Main tabs typically, are unique for each block, therefore, a link to an example of a Main tab is provided. There may be a slight variation between the example and the actual tab.
- N/A = tab is not available for block's configuration form
- No example = no example of tab in document

Tab name	Block names - two different tab sets	
	• **REQFLAGARRAY** • **REQNUMARRAY** • **REQTEXTARRAY**	• **RSPFLAGARRAY** • **RSPNUMARRAY** • **RSPTEXTARRAY**
Main	Click here	Click here
Communications	No example	N/A
Status/Data	Click here	Click here
Identification	Click here	Click here
Dependencies	Click here	Click here
Block Pins	Click here	Click here
Configuration Parameters	Click here	Click here
Monitoring Parameters	Click here	Click here
Block Preferences	Click here	Click here
Template Defining	Click here	Click here

PULSEINPUT

Available library blocks and tabs

The following table lists:

- The individual block names
- The tabs available for each block's Configuration form
- A link to an example of each tab.
  Note: Main tabs typically, are unique for each block, therefore, a link to an example of a Main tab is provided. There may be a slight variation between the example and the actual tab.

Tab name	Block names - all having the same named tabs
	• **PICFASTCUTOFF** • **PICHANNEL** • **PITOTALIZER**
Main	Click here
Identification	Click here
Dependencies	Click here
Block Pins	Click here
Configuration Parameters	Click here
Monitoring Parameters	Click here
Block Preferences	Click here
Template Defining	Click here

Tab name	Block names - all having the same named tabs
	TC-MDP081
Main	Click here

Module Configuration	Click here
Channel Configuration	Click here
Status/Data	Click here
Version	Click here
Server History	Click here
Server Displays	Click here
Control Confirmation	Click here
Identification	Click here

RAIL_IO_HAZ

Available library blocks and tabs

The following table lists:

- The individual block names
- The tabs available for each block's Configuration form
- A link to an example of each tab.
  Note: Main tabs typically, are unique for each block, therefore, a link to an example of a Main tab is provided. There may be a slight variation between the example and the actual tab.

Tab name	Block names - all having the same-named tabs	
	• **TC-PIA082** • **TC-PIB161** • **TC-PIL081**	• **TC-POA081** • **TC-POB041**
Main	Click here	
Module Configuration	Click here	
Channel Configuration	Click here	
Status/Data	Click here	
Version	Click here	
Server History	Click here	
Server Displays	Click here	
Control Confirmation	Click here	
Identification	Click here	

QIMPACT

Available library blocks and tabs

The following table lists:

- The individual block names
- The tabs available for each block's Configuration form
- A link to an example of each tab.
  Note: Main tabs typically, are unique for each block, therefore, a link to an example of a Main tab is provided. There may be a slight variation between the example and the actual tab.

Tab name	Block names - all having the same-named tabs	
	• **QIPACTERM**	• **QIPACCHAN**
Main	Click here	
Module Configuration	Click here	
Channel Data	No example	
Status 1	No example	
Status 2	No example	
QVCS	Click here	
Server History	Click here	
Server Displays	Click here	
Control Confirmation	Click here	
Identification	Click here	

JAGXTREME

Available library blocks and tabs

The following table lists:

- The individual block names
- The tabs available for each block's Configuration form
- A link to an example of each tab.
  Note: Main tabs typically, are unique for each block, therefore, a link to an example of a Main tab is provided. There may be a slight variation between the example and the actual tab.

Tab name	Block names - all having the same-named tabs	
	• **JAGXTERM**	• **JAGXCHAN**
Main	Click here	
Module Configuration	Click here	
Channel Data	No example	
QVCS	Click here	
Server History	Click here	
Server Displays	Click here	
Control Confirmation	Click here	
Identification	Click here	

Configuring modules

Topic	Link
Review the Main tab	Click here
Review the Module Configuration tab	Click here
Review the Channel Configuration tab	Click here
Review the Server History tab	Click here
Review the Server Displays tab	Click here
Review the Diagnostsic Configuration tab	Click here
Review the Status/Data tab	Click here
Review the Version tab	Click here
Review the Control Configuration tab	Click here
Review the Identification tab	Click here
Configure Chassis Analog I/O Module	Click here
Configure Chassis Digital I/O Module	Click here
Configure Chassis Diagnostic I/O Module	Click here
Configure the Serial Interface Module	Click here
Configure the Pulse Input Module	Click here

Defining Series C Function Blocks

This document contains the following Series C operational information for creating Series C function blocks

- Creating Series C I/O AI-HART
- Creating Series C I/O AI-LLMUX
- Creating Series C I/O AO-HART
- Creating Series C I/O DI-24
- Creating Series C I/O DI-HV
- Creating Series C I/O DO-24B

For the complete Series C family architecture refer to the following documents:

- C300 Controller User's Guide - Provides planning and designing activities, as well as the installation, operation, and troubleshooting information for the Series C300 Controller.
- Series C I/O User's Guide - Provides planning, configuration, and troubleshooting information for the Series C I/O Modules.
- Series C Fieldbus Interface Module User's Guide - Provides planning and implementation information for the Series C Fieldbus Interface Module.

Configuring modules - Main tab

The following configuration information pertains to the Main tab for the following modules:

- Chassis Analog I/O
- Chassis Digital I/O
- Chassis Diagnostic I/O
- Serial Interface Module
- Pulse Input Module

Prerequisites:

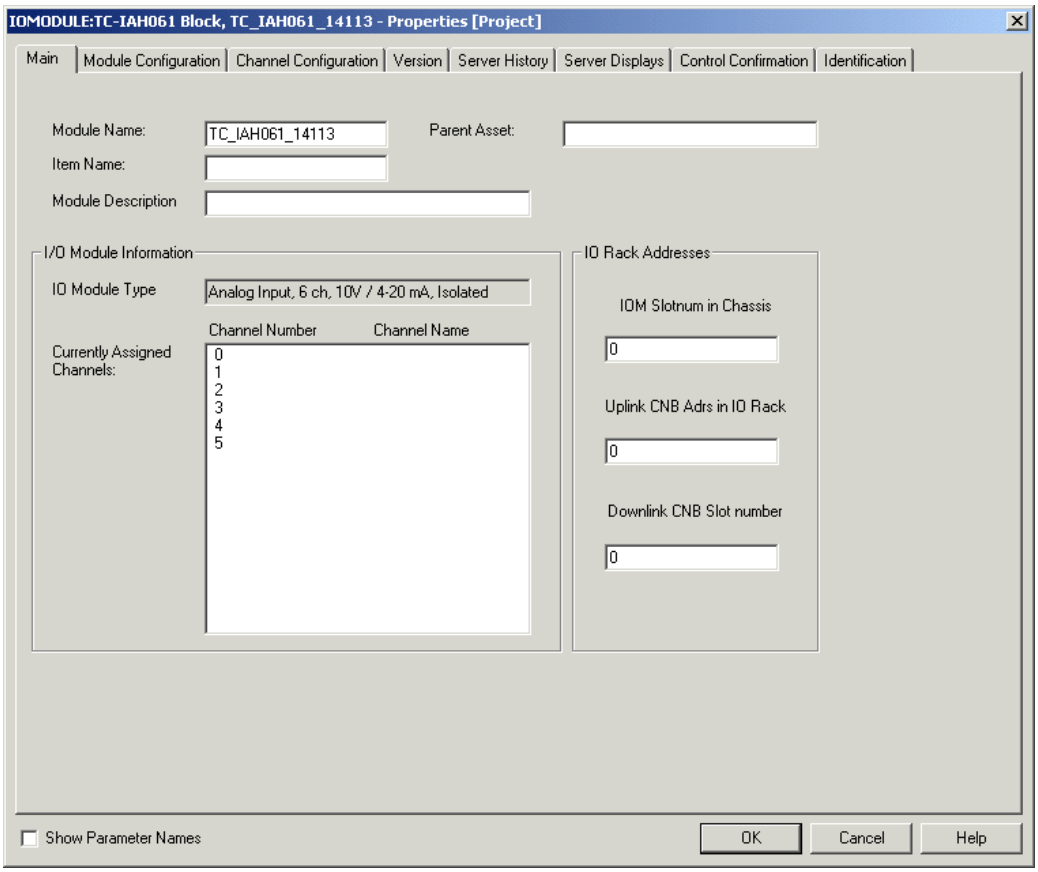
- Control Builder is running

- A control module was created

Considerations:

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

The following is an example of a Configuration form - Main tab.



The following table defines the steps to configure the Main tab.

Step	Action
1	Under the Main tab, enter a Module Name that is more meaningful to you than its default preassigned number (see table below).
2	Parent Asset name is based on relationship established in Enterprise Builder
3	Item name is based on relationship established in Enterprise Builder
4	Enter an optional Module Description to explain the IO Module's function
5	Enter appropriate values for IOM Slotnum in Chassis, Uplink Adrs in I/O Rack and Downlink CNBSlot Number. If necessary, press F1 to access on-line help for assistance during this step.
6	Proceed to the following procedures to configure parameters on the remaining tabs for the module, or click OK to accept only the changes made so far and return to the Project tree.

The following table defines:

- the default preassigned number (that should be changed to a more meaningful name).
- optional module description field.

Configuration form	Default module number (revise to more meaningful name)	Optional module description
Chassis Analog I/O	TC-IAH061xxx (xxx)	Test Strategy AI Module
Chassis Digital I/O	TC-IDA161xxx	Test Strategy DI Module
Chassis Diagnostic I/O	TC-IDX081xxx	Test Diagnostic Module
Serial Interface Module	TC-MUX021xxx	Serial Interface 01
Pulse Input Module	TC-MDP081xxx	Test Strategy PIM

Configuring modules - Module Configuration tab

The following configuration information pertains to the Module Configration tab for the following modules:

- Chassis Analog I/O
- Chassis Digital I/O
- Chassis Diagnostic I/O
- Serial Interface Module
- Pulse Input Module

Prerequisites:

- Control Builder is running
- A control module was created

Considerations:

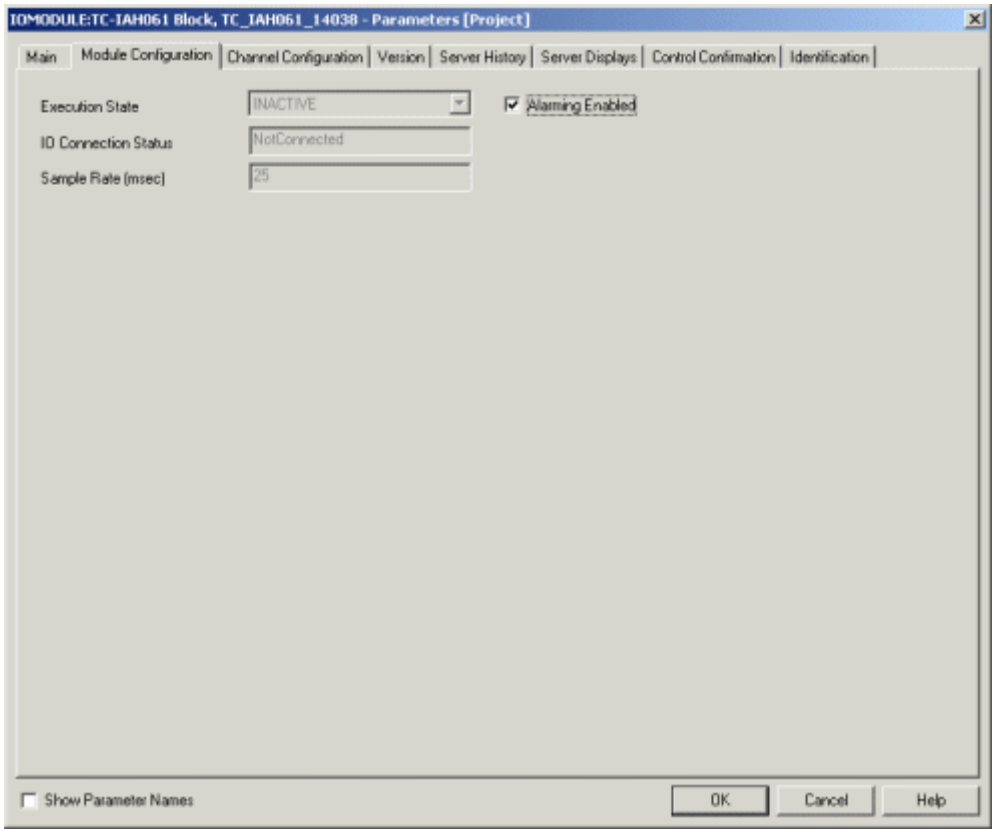
- All illustrations used in the procedure are for example purposes only.



ATTENTION

The only available choice is whether or not to select the Alarming Enabled checkbox. Once the module has been loaded and activated, additional parameters (such as Execution State) may be configured on the Monitoring Tree window. If necessary, press F1 to access on-line help for assistance during this step.

The following is an example of a Configuration form - Module Configuration tab.



The following table defines the steps to configure the Module Configuration tab.

Step	Action
1	Under the Module Configuration tab, enter desired values for configuration parameters.
2	Proceed to the following procedures to configure parameters on the remaining tabs for the module, or click OK to accept only the changes made so far and return to the Project tree.

Configuring modules - Channel Configuration tab

The following configuration information pertains to the Channel Configuration tab for the following modules:

- Chassis Analog I/O
- Chassis Digital I/O
- Chassis Diagnostic I/O
- Serial Interface Module
- Pulse Input Module

Prerequisites:

- Control Builder is running
- A control module was created

Considerations:

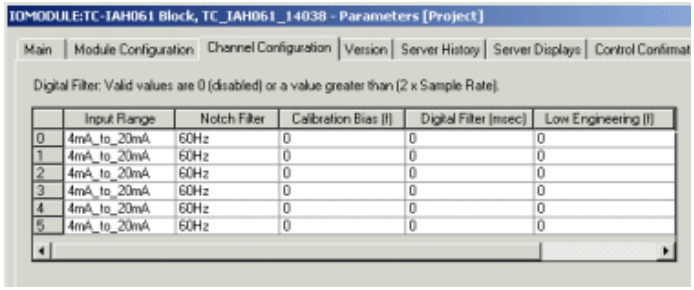
- All illustrations used in the procedure are for example purposes only.

The following table defines the steps to configure the Channel Configuration tab.

Step	Action
1	Under the Channel Configuration tab, enter desired values for configuration parameters.
2	Proceed to the following procedures to configure parameters on the remaining tabs for the module, or click OK to accept only the changes made so far and return to the Project tree.

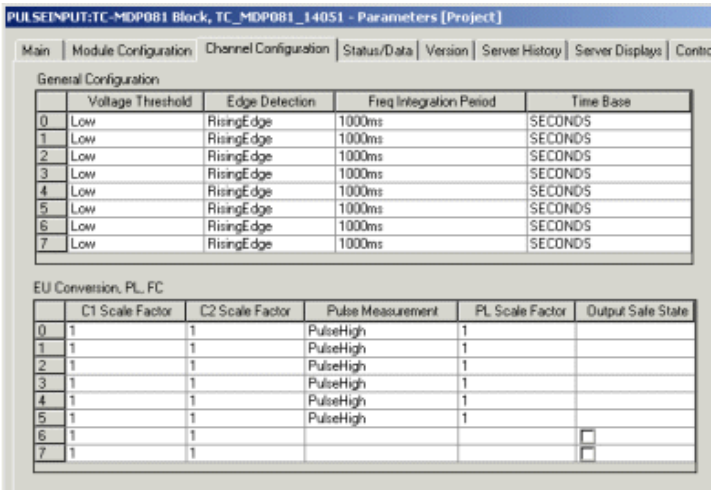
The following table defines the channel configuration settings which are available for the various modules.

Channel Configration parameter settings
Chassis Analog I/O Configuration parameters vary depending upon the type of Analog I/O Module being configured. If necessary, press F1 to access on-line help for assistance during this step.



Pulse Input Module

Configuration parameters are set for the fields shown in the graphic below. If necessary, press F1 to access on-line help for assistance during this step.



Chassis Digital I/O

Chassis Diagnostic I/O

Serial Interface Module

There is no Channel Configuration Data for the modules.



Configuring modules - Server History tab

The following configuration information pertains to the Server History tab for the following modules:

- Chassis Analog I/O
- Chassis Digital I/O
- Chassis Diagnostic I/O
- Serial Interface Module
- Pulse Input Module

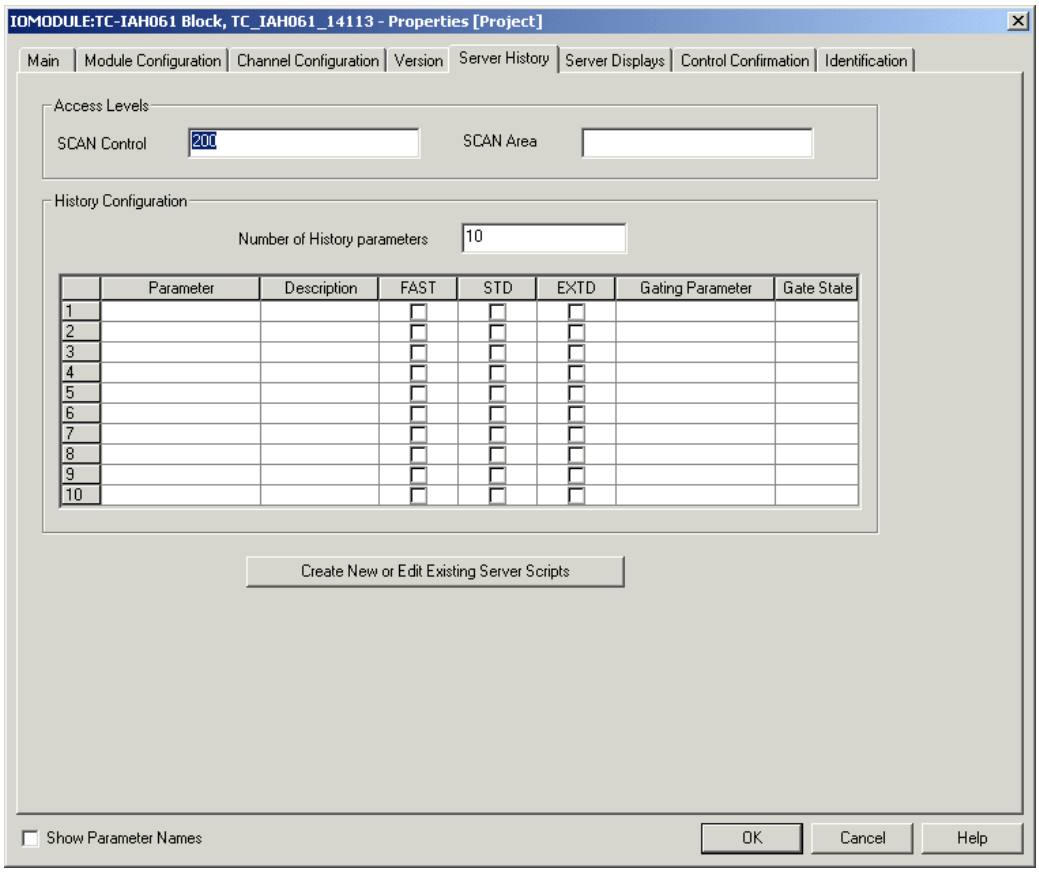
Prerequisites:

- Control Builder is running
- A control module was created

Considerations:

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

The following is an example of a Configuration form - Server History tab.



The following table defines the steps to configure the Server History tab.

Step	Action
1	Under the Server History tab, enter the appropriate information for SCAN Control and SCAN Area along with values for appropriate parameters related to history collection and archiving. If necessary, press F1 to access on-line help for assistance during this step
2	Proceed to the following procedures to configure parameters on the remaining tabs for the Analog I/O Module, or click OK to accept only the changes made so far and return to the Project tree.

Configuring modules - Server Displays tab

The following configuration information pertains to the Server Displays tab for the following modules:

- Chassis Analog I/O
- Chassis Digital I/O
- Chassis Diagnostic I/O
- Serial Interface Module
- Pulse Input Module

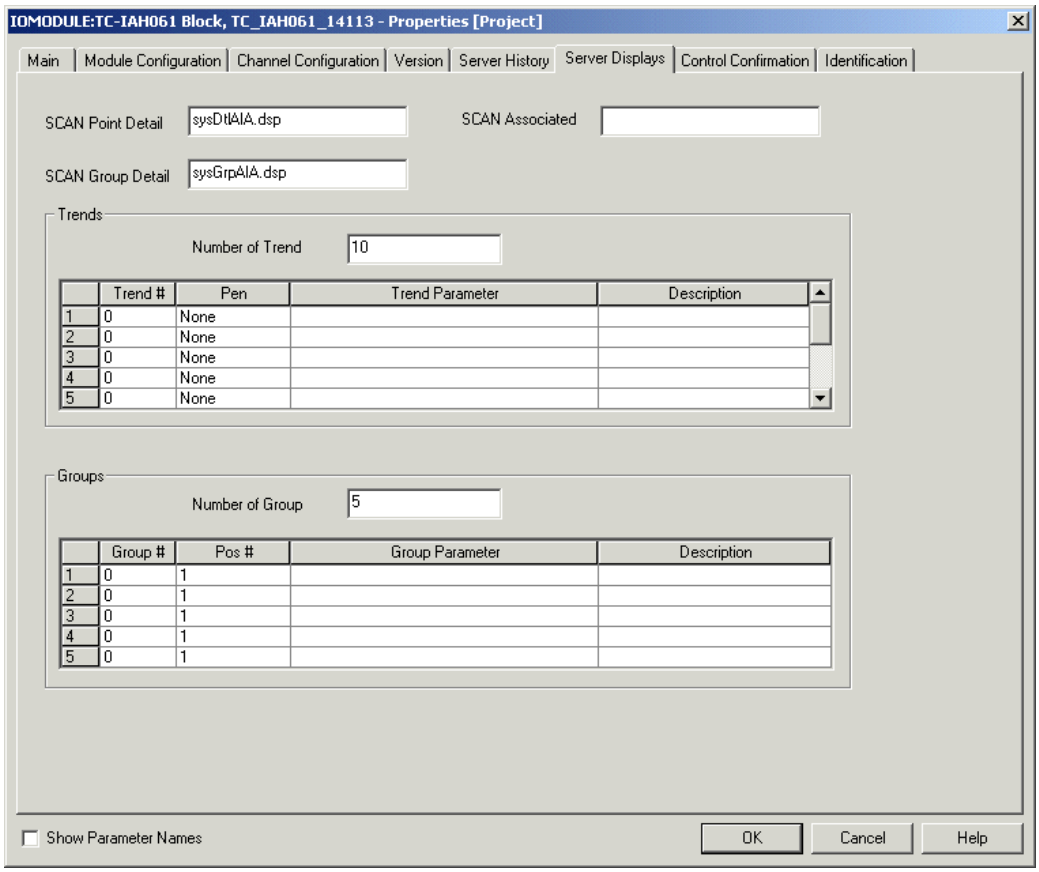
Prerequisites:

- Control Builder is running
- A control module was created

Considerations:

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

The following is an example of a Configuration form - Server Displays tab.



The following table defines the steps to configure the Server Displays tab.

Step	Action
1	Under the Server Displays tab, enter the appropriate information to specify related SCAN Point Detail (display), SCAN Group Detail (display), and SCAN Associated (display) along with values for appropriate parameters to define Trends and Groups for display. If necessary, press F1 to access on-line help for assistance during this step
2	Click OK on the configuration form to accept all configuration selections made on each configuration tab and to return to the Project tree.

Configuring modules - Diagnostic Configuration tab

The following configuration information pertains to the Diagnostic Configuration tab for the following module:

- Chassis Diagnostic I/O

Prerequisites:

- Control Builder is running
- A control module was created

Considerations:

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

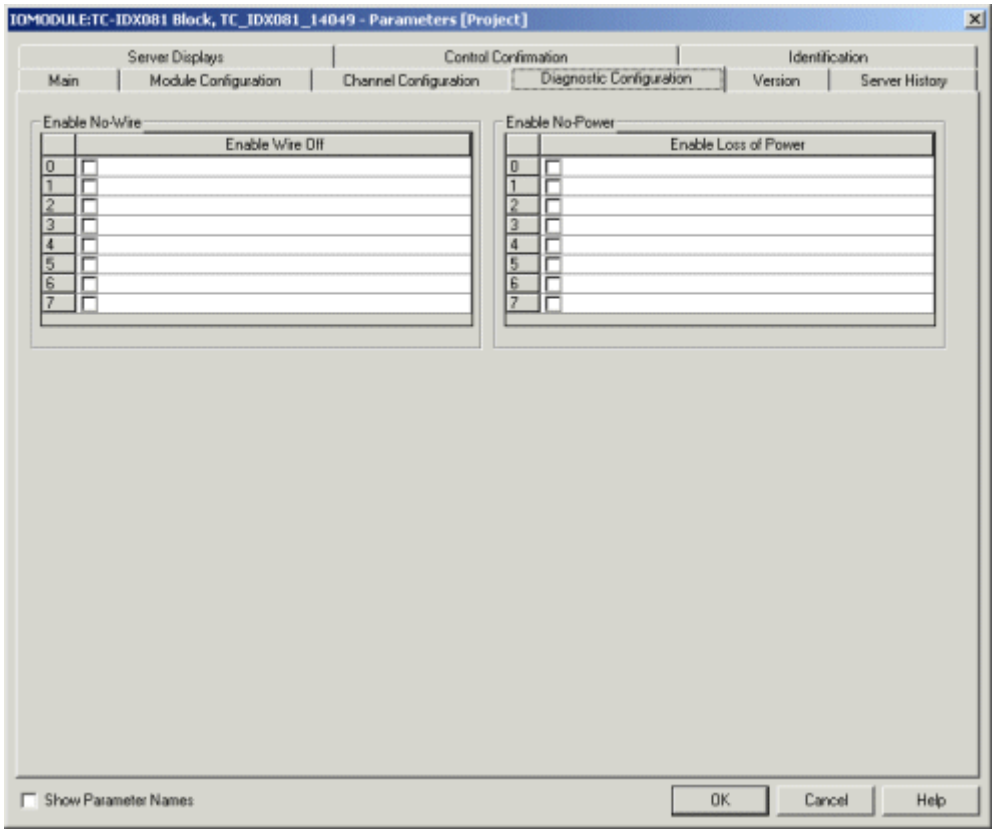
ATTENTION

Configuration parameters vary depending upon the type of module being configured. If necessary, press F1 to access on-line help for assistance during this step.

The following table defines the steps to configure the Diagnostic Configuration tab.

Step	Action
1	Under the Diagnostic Configuration tab (see figure below), enter desired values for configuration parameters.
2	Proceed to the following procedures to configure parameters on the remaining tabs for the Diagnostic I/O Module, or click OK to accept only the changes made so far and return to the Project tree.

The following is an example of a Configuration form - Diagnostic Configuration tab.



Configuring modules - Status/Data tab

The following configuration information pertains to the Status/Data tab for the following module:

- Pulse Input Module

Prerequisites:

- Control Builder is running
- A control module was created

Considerations:

- The Status/Data tab includes data fields that are only active in the Monitoring mode and they are unavailable for configuration in the Project mode.
- All illustrations used in the procedure are for example purposes only.

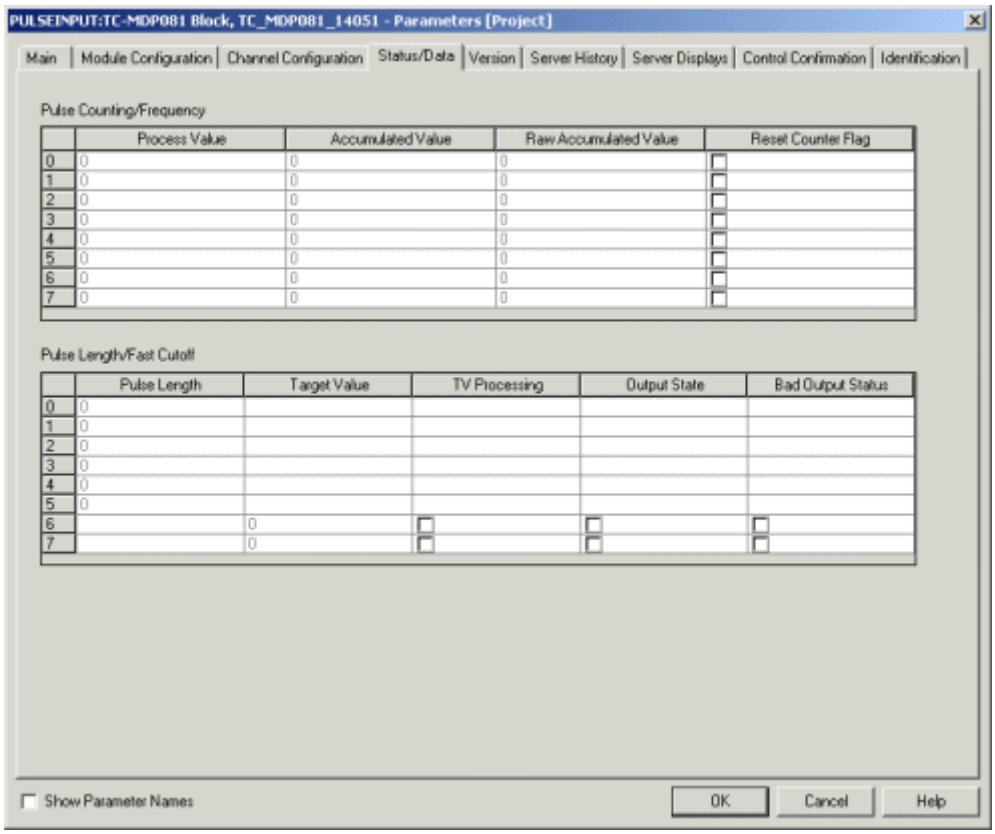
ATTENTION

Configuration parameters vary depending upon the type of module being configured. If necessary, press F1 to access on-line help for assistance during this step.

The following table defines the steps to configure the Diagnostic Configuration tab.

Step	Action
1	Under the Status/Data tab (see figure below), enter desired values for configuration parameters.
2	Proceed to the following procedures to configure parameters on the remaining tabs for the module, or click OK to accept only the changes made so far and return to the Project tree.

The following is an example of a Configuration form - Status/Data tab.



Configuring modules - Version tab

The following configuration information pertains to the Version tab for the following modules:

- Chassis Analog I/O
- Chassis Digital I/O
- Chassis Diagnostic I/O
- Serial Interface Module
- Pulse Input Module

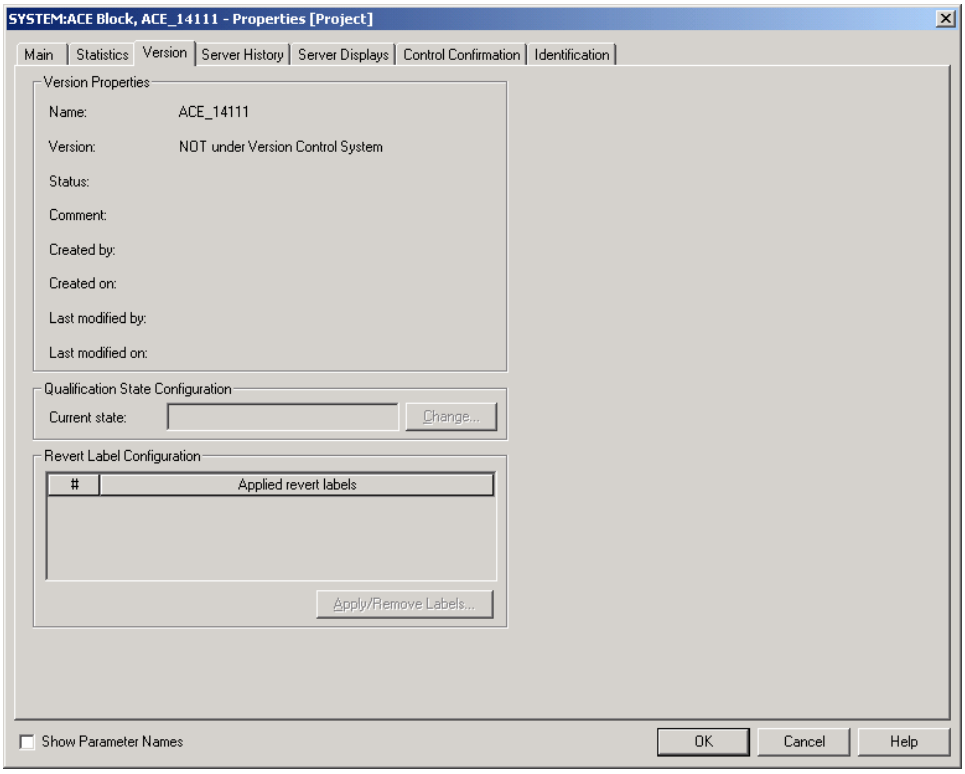
Prerequisites:

- Control Builder is running
- A control module was created

Considerations:

- The Version tab is for reviewing established settings.
- All illustrations used in the procedure are for example purposes only.

The following is an example of a Configuration form - Version tab.



Configuring modules - Control Confirmation tab

The following configuration information pertains to the Control Confirmation tab for the following modules:

- Chassis Analog I/O
- Chassis Digital I/O
- Chassis Diagnostic I/O
- Serial Interface Module
- Pulse Input Module

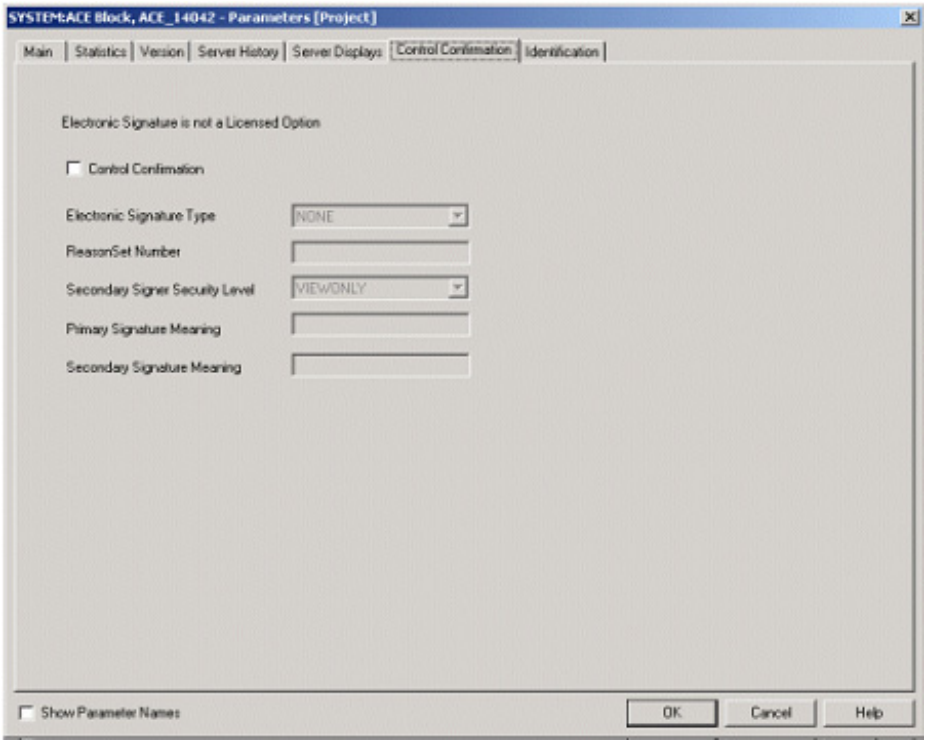
Prerequisites:

- Control Builder is running
- A control module was created

Considerations:

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

The following is an example of a Configuration form - Control Confirmation tab.


SYSTEMACE Block, ACE_14042 - Parameters [Project]
Main
Statistics
Version
Server History
Server Displays
Control Confirmation
Identification
Electronic Signature is not a Licensed Option
Control Confirmation
Electronic Signature Type
NONE
ReasonSet Number
Secondary Signer Security Level
VIEWONLY
Primary Signature Meaning
Secondary Signature Meaning
Show Parameter Names
OK
Cancel
Help

Configuring modules - Identifcation tab

The following configuration information pertains to the Identification tab for the following modules:

- Chassis Analog I/O
- Chassis Digital I/O
- Chassis Diagnostic I/O
- Serial Interface Module
- Pulse Input Module

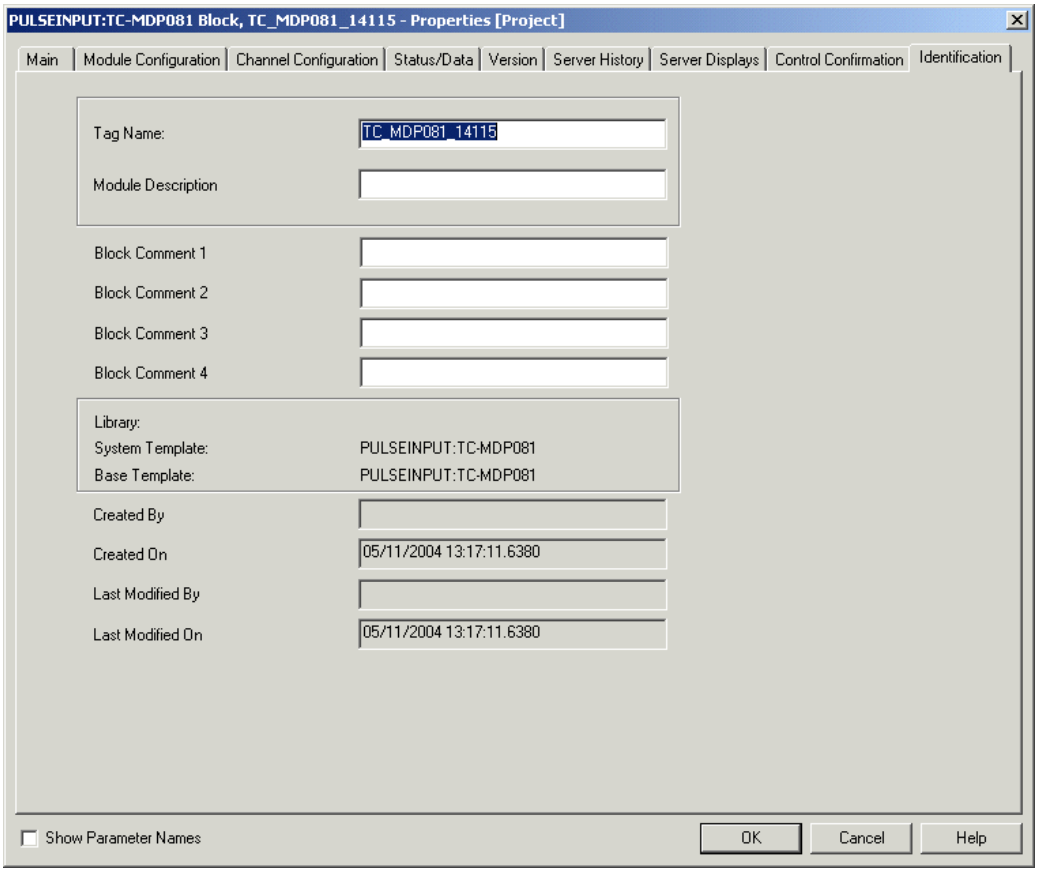
Prerequisites:

- Control Builder is running
- A control module was created

Considerations:

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

The following is an example of a Configuration form - Identification tab and allows user comments to be added.


PULSEINPUT:TC-MDP081 Block, TC_MDP081_14115 - Properties [Project]
Main
Module Configuration
Channel Configuration
Status/Data
Version
Server History
Server Displays
Control Confirmation
Identification
Tag Name:
TC_MDP081_14115
Module Description
Block Comment 1
Block Comment 2
Block Comment 3
Block Comment 4
Library:
System Template:
PULSEINPUT:TC-MDP081
Base Template:
PULSEINPUT:TC-MDP081
Created By
Created On
05/11/2004 13:17:11.6380
Last Modified By
Last Modified On
05/11/2004 13:17:11.6380
Show Parameter Names
OK
Cancel
Help

Configuring Chassis Analog I/O module

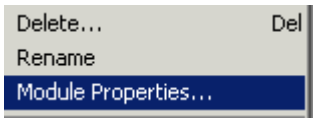
Use the following procedure to customize a Chassis Analog I/O Module (either AI Module such as model TC-IAH061 or an AO Module such as model TC-OAH061).

Prerequisites:

- Control Builder is running
- A control module was created

Considerations:

- You can customize an I/O Module by changing its name, description, IOM Slotnum in Chassis, UPLINK CNB MAC ID, as well as other configuration parameters with regards to Module, Channel and Server configuration. While this procedure is specific to Chassis type I/O Modules, it can easily be adapted to other supported I/O types.

Step	Action	Result
1	Right-click the desired IOM block icon in the Project tree window.	Calls up the shortcut menu. 
2	Click Module Properties...	Calls up the IOM's configuration form.
3	Go to the following table and configure the parameters using the configuration form tabs.	Enter appropriate IOM data into the Engineering Repository Database (ERDB).

Configuring the module by using the configuration form tabs

Step	Configuration form tab	See
1	Main tab	Click here
2	Module Configuration tab	Click here
3	Channel Configuration tab	Click here
4	Version tab	Click here
5	Server History tab	Click here
6	Server Displays tab	Click here
7	Control Confirmation tab	Click here
8	Identification tab	Click here

Configuring Chassis Digital I/O module

Use the following procedure to customize a Digital I/O Module (either a DI Module such as TC-IDA161 or a DO Module such as TC-ODA161).

Prerequisites:

- Control Builder is running
- A control module was created

Considerations:

- You can customize an I/O Module by changing name, description, IOM Slotnum in Chassis, UPLINK CNB MAC ID,, as well as other configuration parameters with regards to Module, Channel and Server configuration. While this procedure is specific to Chassis type I/O Modules, it can easily be adapted to other supported I/O types.
- All illustrations used in the procedure are for example purposes only.

Step	Action	Result
1	Right-click the desired IOM block icon in the Project tree window.	Calls up the Context Menu. 
2	Click Module Properties...	Calls up the IOM's configuration form.
3	Go to the following table and configure the parameters using the configuration form tabs.	Enter appropriate IOM data into the Engineering Repository Database (ERDB).

Configuring the module by using the configuration form tabs

Step	Configuration form tab	See
1	Main tab	Click here
2	Module Configuration tab	Click here
3	Channel Configuration tab	Click here
4	Version tab	Click here
5	Server History tab	Click here
6	Server Displays tab	Click here
7	Control Confirmation tab	Click here
8	Identification tab	Click here

Configuring Chassis Diagnostic I/O Module

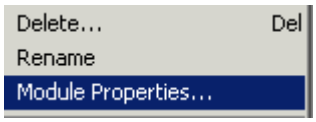
Use the following procedure to customize a Diagnostic I/O Module (either a digital input diagnostic module such as TC-IDX081 or a digital output diagnostic module such as TC-ODX081).

Prerequisites:

- Control Builder is running
- A control module was created

Considerations:

- You can customize an I/O Module by changing its name, description, IOM Slotnum in Chassis, UPLINK CNB MAC ID, as well as other configuration parameters with regards to Module, Channel and Server configuration. While this procedure is specific to Chassis type I/O Modules, it can easily be adapted to other supported I/O types.
- All illustrations used in the procedure are for example purposes only.

Step	Action	Result
1	Right-click the desired IOM block icon in the Project tree window.	Calls up the Context Menu. 
2	Click Module Properties...	Calls up the IOM's configuration form.
3	Go to the following table and configure the parameters using the configuration form tabs.	Enter appropriate IOM data into the Engineering Repository Database (ERDB).

Configuring the module by using the configuration form tabs

Step	Configuration form tab	See
1	Main tab	Click here
2	Module Configuration tab	Click here
3	Channel Configuration tab	Click here
4	Diagnostic Configuration tab	Click here
5	Status/Data tab	Click here
6	Version tab	Click here
7	Server History tab	Click here
8	Server Displays tab	Click here
9	Control Confirmation tab	Click here
10	Identification tab	Click here

Configuring the Serial Interface Module

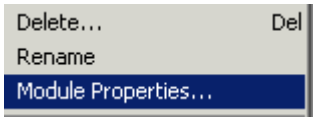
Perform the following steps to customize a Serial Interface Module (such as, TC-MUX021).

Prerequisites:

- Control Builder is running
- A control module was created

Considerations:

- You can customize the Serial Interface Module by changing its name, description, IOM Slotnum in Chassis, UPLINK CNB MAC ID,, as well as other configuration parameters with regards to Module, Channel and Server configuration.
- All illustrations used in the procedure are for example purposes only.

Step	Action	Result
1	Right-click the desired IOM block icon in the Project tree window.	Calls up the Context Menu. 
2	Click Module Properties...	Calls up the IOM's configuration form.
3	Go to the following table and configure the parameters using the configuration form tabs.	Enter appropriate IOM data into the Engineering Repository Database (ERDB).

Configuring the module by using the configuration form tabs

Step	Configuration form tab	See
1	Main tab	Click here
2	Module Configuration tab	Click here
3	Channel Configuration tab	Click here
4	Version tab	Click here
5	Server History tab	Click here
6	Server Displays tab	Click here
7	Control Confirmation tab	Click here
8	Identification tab	Click here

Configuring the Pulse Input Module

Perform the following steps to customize a Pulse Input Module (such as, TC-MDP081).

Prerequisites:

- Control Builder is running
- A control module was created

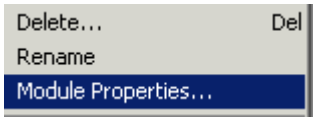
Considerations:

ATTENTION

When configuring the Pulse Input Module:

- Use the Pulse Input Channel block (PICHANNEL in the Control Builder library) to configure Channels 0 through 5 of the pulse input module.
- Use the Pulse Input Channel with Fast Cutoff block (PICFASTCUTOFF in the Control Builder Library) to configure channels 6 and 7.
- For additional details, see Pulse Input Modulein the Control *Component Library (CCL) Help*.

- You can customize the Pulse Input Module by changing its name, description, IOM Slotnum in Chassis, UPLINK CNB MAC ID, as well as other configuration parameters with regards to Module, Channel and Server configuration.
- All illustrations used in the procedure are for example purposes only.

Step	Action	Result
1	Right-click the desired IOM block icon in the Project tree window.	Calls up the Context Menu. 
2	Click Module Properties...	Calls up the IOM's configuration form.
3	Go to the following table and configure the parameters using the	Enter appropriate IOM data into the Engineering Repository Database

Step	Action	Result
	configuration form tabs.	(ERDB).

Configuring the module by using the configuration form tabs

Step	Configuration form tab	See
1	Main tab	Click here
2	Module Configuration tab	Click here
3	Channel Configuration tab	Click here
4	Status/Data tab	Click here
5	QVCS tab	Click here
6	Server History tab	Click here
7	Server Displays tab	Click here
8	Control Confirmation tab	Click here
9	Identification tab	Click here

Control Builder Operations

Creating hardware modules

This section includes information that you can reference for:

Topic	Link
Create a CPM and CEE	Click here
Create an ACE and CEE	Click here
Create an External Server	Click here
Create an Inter Cluster Gateway	Click here
Create an IOLOM and IOLINK	Click here
Create a Redundancy Module	Click here
Create an instance of Input/Output Modlue	Click here
Create a strategy to use insertion points	Click here

Creating a CPM and CEE

Use the following procedure to create a Control Processor Module (CPM) and associated Control Execution Environment (CEE) blocks in the Project Tree that will represent an installed C200 CPM. The CEE supports execution of a set of Function Blocks for solving control applications. It runs in the CPM as a software layer built on top of the control software infrastructure.



TIP

You can configure a CPM block in the Control Builder Project tab without the CPM hardware being installed. However, it is good idea to have the communications driver and hardware that is going to be used for the system installed, configured, and running. The CPM needs the name of the communications driver specified on its configuration form to complete its configuration data. The CPM represents a hardware module and the block configuration specifies the communication path to the hardware.

Configuring the Main tab

The following procedure is used for configuring the Main tab for a CPM and CEE.

Prerequisites:

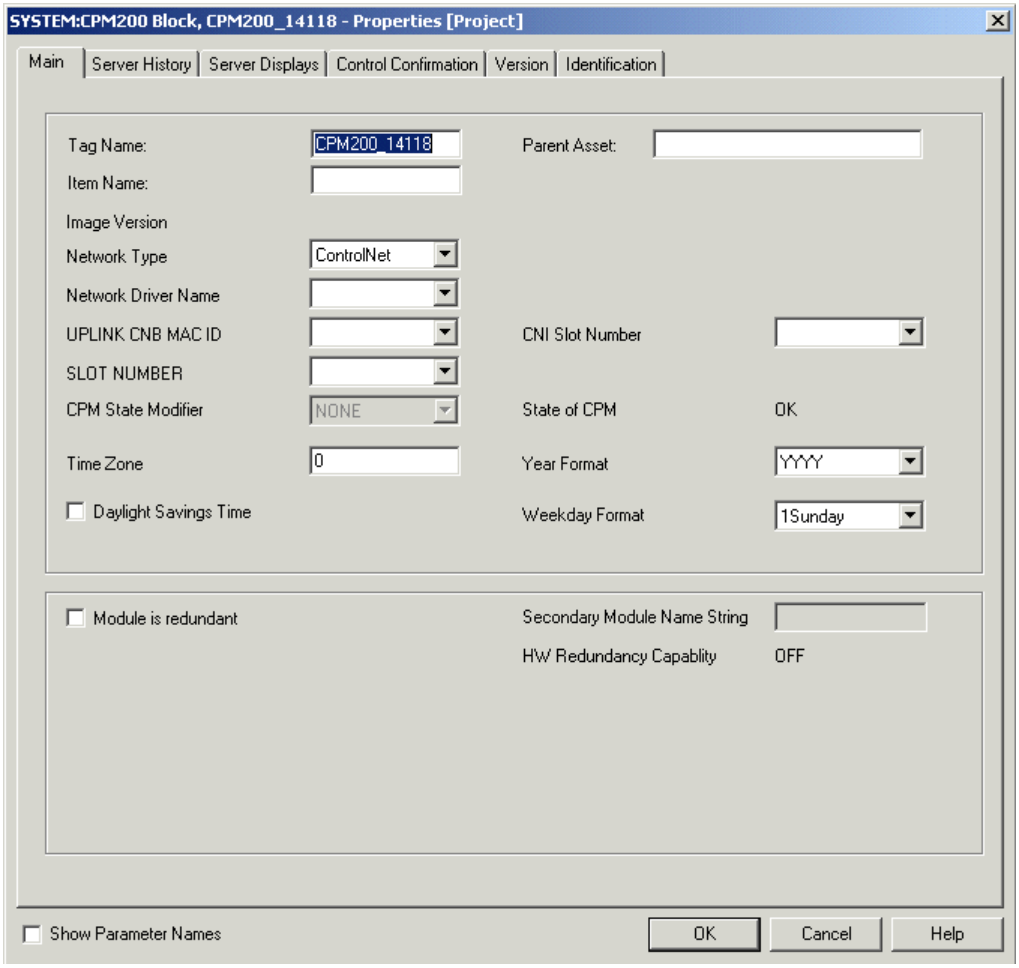
- Control Builder is running
- Tree windows are open

Considerations:

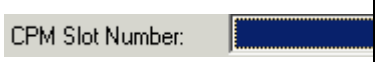
- All illustrations used in the procedure are for example purposes only.

Step	Action	Result
1	Click File -> New -> Controllers -> CPM200 - Control Processor.	Calls up the CPM200 Block configuration form with Tag Name field highlighted.

Step	Action	Result

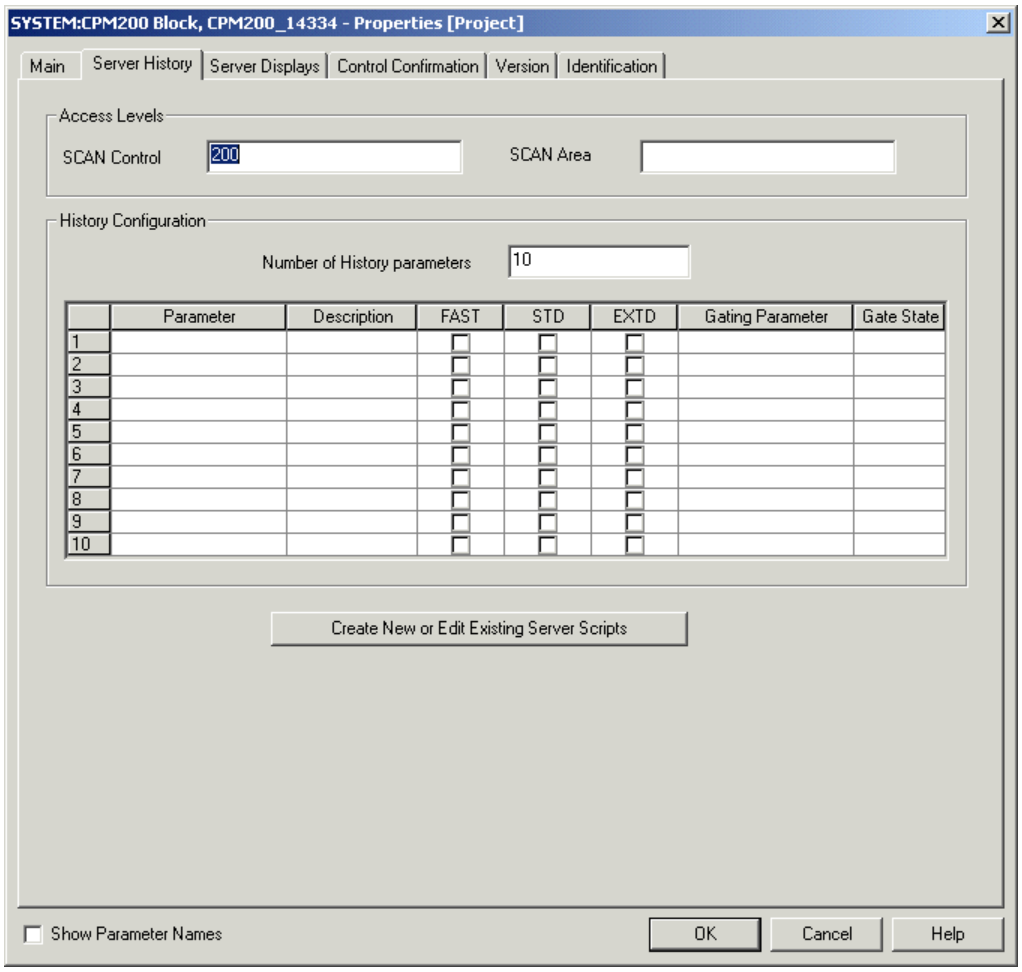


2	Key in desired name of up to 16 characters or accept the default. Press <Tab>.	Moves cursor to Network Type field. 

Step	Action	Result
	TIP All communications drivers must be installed and running before they will appear in the Driver Name selection list.	
3	Accept default or click down-arrow button and select from list. Press <Tab>.	Moves cursor to Network Driver Name field. 
4	Click down-arrow button and select installed ControlNet or Ethernet driver from the list. Press <Tab>.	Moves cursor to UPLINK CNB MAC ID field. 
5	Click down-arrow button and select Address of the CNI (ControlNet Interface) card in the chassis that is connected to the communications adapter card in the server. Press <Tab>.	Moves cursor to Slot Number field. 
6	Click down-arrow button and select slot number where supervisory CNI card is installed in the chassis. CNI card takes up one slot in the chassis and slots are numbered from left starting at 0. Press <Tab>.	Moves cursor to Time Zone field. 
7	Key in the appropriate time zone offset value for the location where the CPM is installed. Press <Tab>.	Moves cursor to Daylight Savings Time. 
8	Leave box unchecked, if Daylight Savings Time is not currently being observed at your location. Or, Check the box, if Daylight Savings time is currently being observed at your location. Press <Tab>.	Moves cursor to CPM Slot Number field. (The Command field is not accessible in Project mode.) 

Step	Action	Result
9	Click down-arrow button and select slot number where CPM is installed in the chassis from the list. CPM takes up 2 slots in the chassis and slots are numbered from left starting at 0. Only odd-numbered slot is selected. Press <Tab>.	Moves cursor to the Year Format field. 
10	Accept default or click down-arrow button and select desired format from the list. Press <Tab>.	Moves cursor to Weekday Format field. 
	TIP The time zone represents the offset value from the Greenwich Mean Time (GMT) based on your geographical location. For example, the time zone value for a CPM located in the Eastern time zone of the United States that is currently not observing daylight savings time would be -05.0 or -5. Always use the offset value that is not adjusted for daylight savings time as the entry for the Time Zone field. For example, the adjusted offset value for the Eastern time zone of the United States is -04.0, but use the unadjusted value of -5 instead. You may want to visit the http://www.worldtimeserver.com/ website, if you have a question about the appropriate offset value for your given location.	
11	Press <Tab>.	Moves cursor to Module is Redundant check box.
12	Leave the box unchecked, if non-redundant CPM configuration is intended. Check the box, if CPM is part of a Redundant Chassis Pair (RCP). Press <Tab>.	If the box is unchecked, cursor moves to Year Format field. Go to Step 15. If the box is checked, cursor moves to Secondary Module Name String field. 
13	Enter Secondary Module Name String name. Press <Tab>.	Moves cursor to Item Name field.
14	Enter Item Name. Press <Tab>.	Moves cursor to Parent Asset field.
15	Enter Parent Asset name.	

Step	Action	Result
16	• Click Server History tab.	Calls up the Server History configuration form.



Configuring the Server Displays tab

The following procedure is used for configuring the Server Displays tab for CPM or ACE.

Prerequisites:

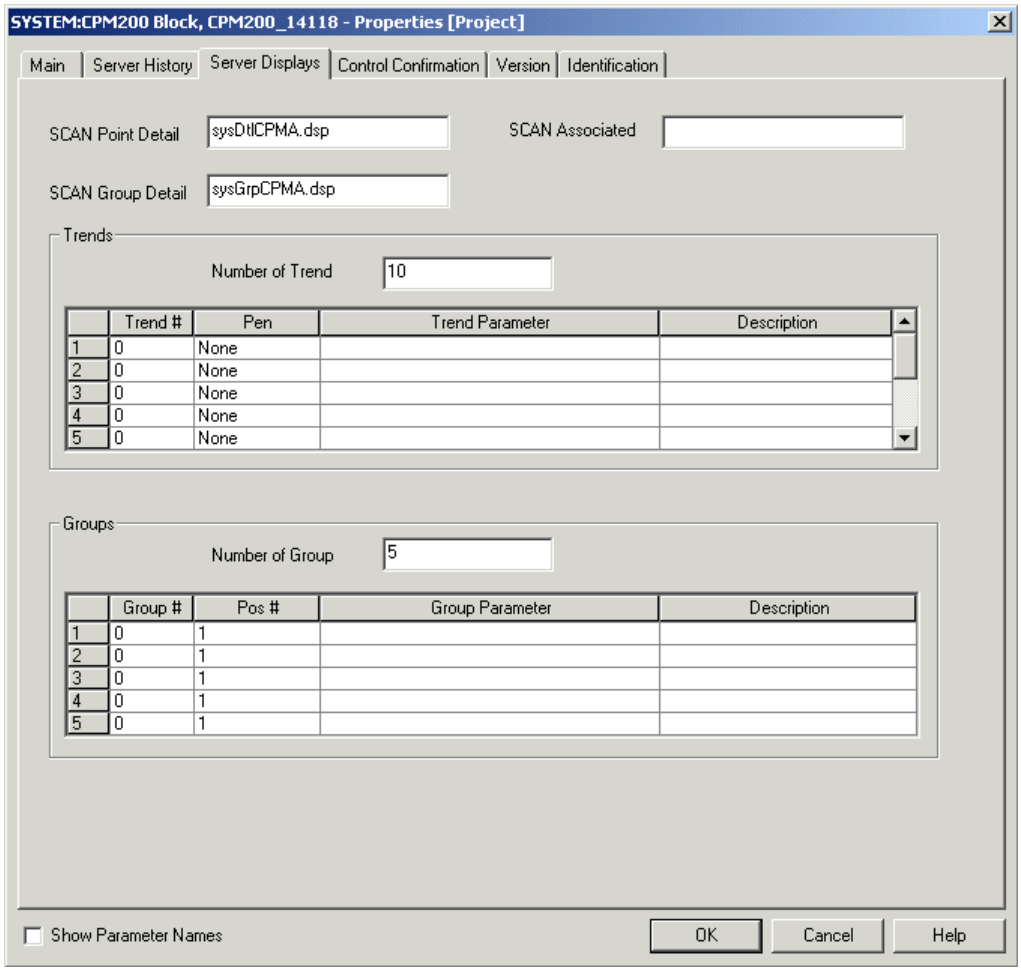
- Control Builder is running
- Tree windows are open

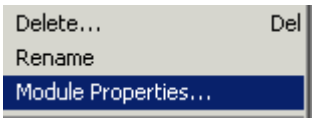
Considerations:

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

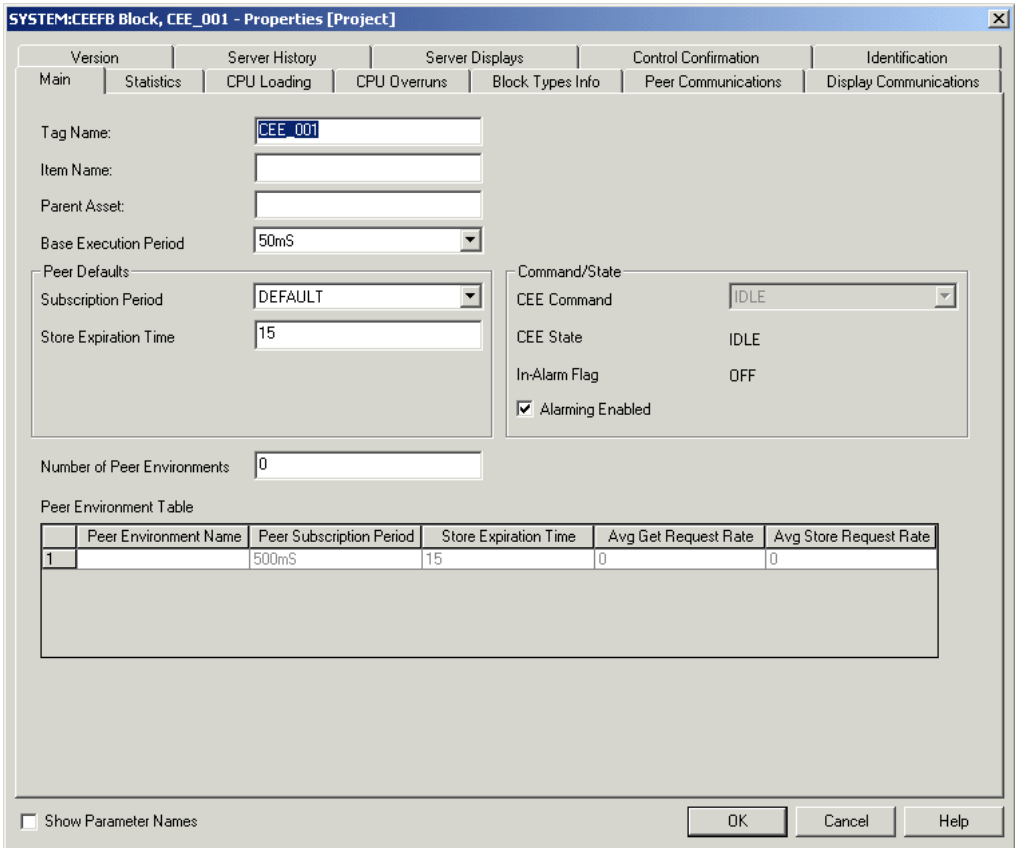
Step	Action	Result
	**TIP** If you have a Distributed Server Architecture (DSA), you must enter the Control Area assignment for this Server. (Note that area code assignments are made through the Station application.) If you do not have a DSA, you can skip this field if area is not enabled through the Station application.	
1	Use the on-line help as a guide to complete the configuration entries on this tab. Click the Server Displays tab.	Calls up the Server Displays configuration form.

Step	Action	Result

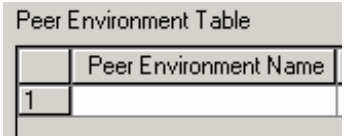


Step	Action	Result
2	Use the on-line help as a guide to complete the configuration entries on this tab. Click the OK button.	CPM Closes the form and creates CPM/CEE block icons in Project tab. ACE Closes the form and creates ACE/CEE block icons in Project tab. 
3	CPM Right-click CEE block icon. ACE Right-click CEEACE block icon.	Calls up shortcut menu. 
4	Click Module Properties...	Calls up CEE Block configuration form.

Step	Action	Result



5	Key in desired name of up to 16 characters or accept the default. Press <Tab>.	Moves cursor to Item Name field
6	Enter Item Name. Press <Tab>.	Moves cursor to Parent Asset field
7	Enter Parent Asset name. Press <Tab>.	Moves cursor to Base Execution Period field. CPM example 

Step	Action	Result
8	Accept the default or click down-arrow button and select desired period. Press <Tab>.	Moves cursor to Subscription Period field.
9	Accept the default or click down-arrow button and select desired period. Press <Tab>.	Moves cursor to Store Expiration Time.
10	Accept the default or key in desired value. Press <Tab>.	Moves cursor to the Number of Peer Environments. Since CEE Command field is unavailable in Project mode and the CEE State field is read only.
	TIP The Number of Peer Envs and Peer Environment Table are interactive. The value entered for the Number of Peer Envs determines how many rows appear in the Peer Environment Table.	
11	Key in number of peer environments for this CPM. Press <Tab>. Or, skip this field, if no peer environments will be used. Click Server History tab.	If peer environments will be used, cursor moves to the Peer Environment Name of column in the Peer Environment Table.  If no peer environments exist, go to Step 13.
12	Key in valid name for existing peer environment. Press <Tab>.	Moves cursor to Peer Subscription Period column. CPM example 

Step	Action	Result
11	Accept default or click down-arrow button to select another value specific to the given environment. Press <Tab>.	Moves cursor to Store Expiration Time column. 
13	Accept the default or key in desired value. Click Server History tab.	Calls up the Server History configuration form.
14	Repeat Steps 1 and 2.	Completes CEE configuration and closes the form.
15	This completes the creation procedure for either: CPM/CEE, or ACE/CEE.	

REFERENCE - INTERNAL

See the Experion Control Builder Components Reference, Component Categories and Types and the Experion Planning Guide, Planning Your ControlNet Addressing for details on setting MAC addresses and determining correct slot numbers for your system.

Creating an ACE and CEE

Use the following procedure to create an Application Control Environment (ACE) controller and associated Control Execution Environment (CEE) blocks in the Project Tree that will represent an installed ACE controller. The CEE supports execution of a set of Function Blocks for solving control applications. It runs in the ACE controller as a software layer built on top of the control software infrastructure.

In ACE/SimACE you cannot directly connect to I/O unless it is a 1756 channel. You will receive a 2299 error stating “initiator cannot establish connection with the responder”. Ensure your connection is defined as a 1756 channel.

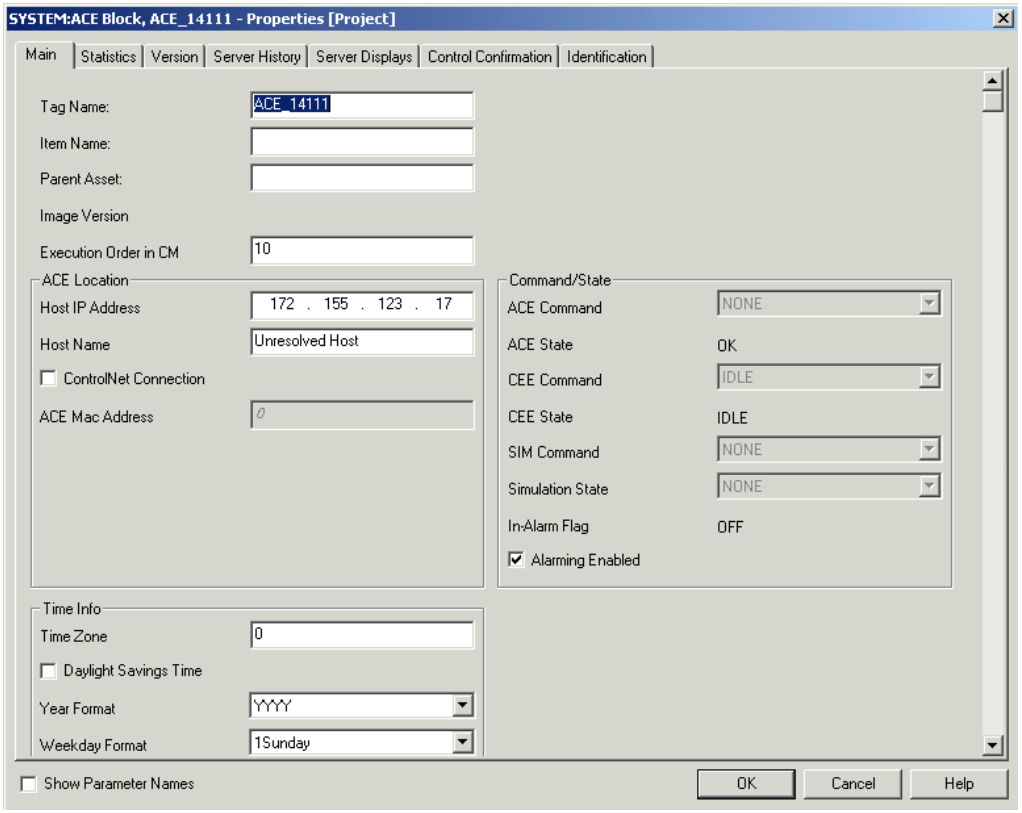
Prerequisites:

- Control Builder is running
- Tree windows are open

Considerations:

- The CEE supports execution of a set of Function Blocks for solving control applications.
- It runs in the ACE supervisory controller as a software layer built on top of the control software infrastructure.
- All illustrations used in the procedure are for example purposes only.

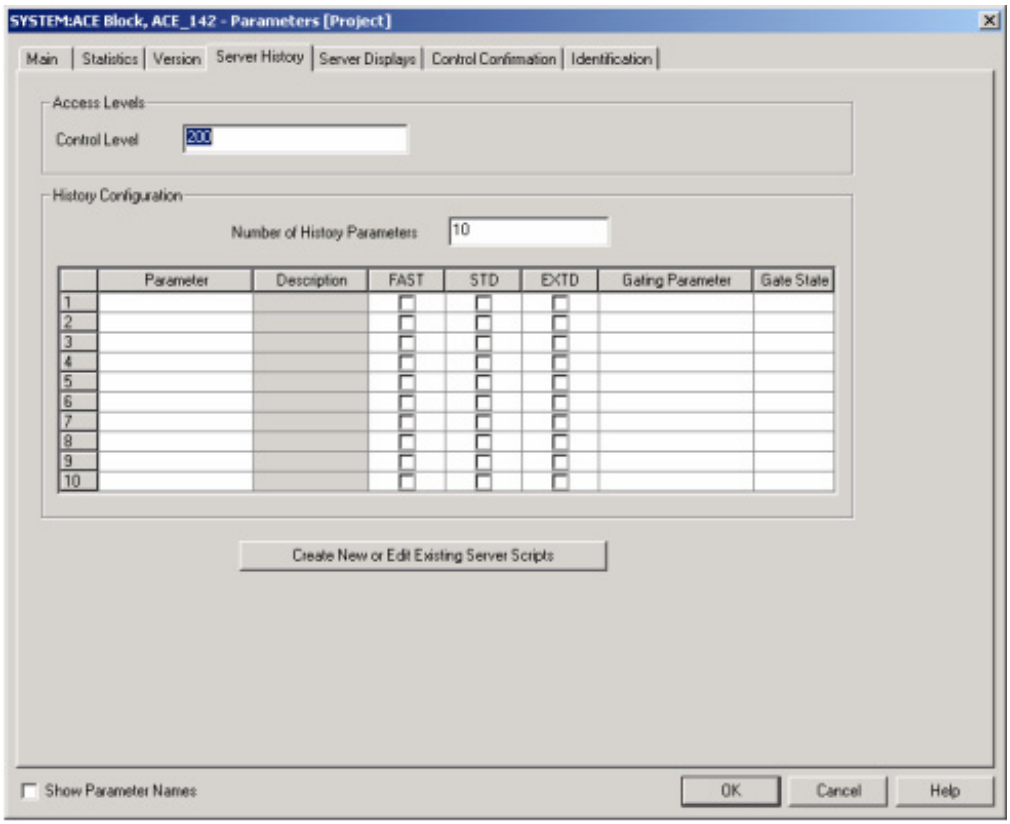
Step	Action	Result
1	Click File -> New -> Controllers -> ACE - Application Control Environment.	Calls up the ACE Block configuration form with Tag Name field highlighted.

Step	Action	Result
		
2	Key in desired name of up to 16 characters or accept the default. Press <Tab>.	Moves the cursor to the Item Name field.
3	If the CEE is part of an Enterprise Model, enter its Item Name here. Press <Tab>	Moves cursor to the Execution Order in CM field.
4	Retain the default execution order of 10, or enter a new value. Press <Tab>.	Moves cursor to Host IP Address field.
	TIP Host name and Host IP Address are interactive entries. We recommend that you key in the Host IP Address first and let the system determine the Host Name automatically.	

Step	Action	Result
	This is especially true if you are configuring the ACE block with the ACE node offline. In this case, entering the: • Host IP address first generates a Warning message, but • entering the Host Name first generates an error message. When keying in an IP address, use the mouse or the left and right arrow keys to move the cursor to locations within the field. Do not press the <Tab> key until the complete address is keyed in.	
5	Key in the host pc IP address for the ACE node. Press <Tab>. Or, press <Tab> to skip this field and enter Host Name instead. Acknowledge any error message prompts.	System automatically determines the Host Name, when ACE node is online, and moves cursor to Host Name field.  (Valid IP address entry results in system automatically determining the Host Name, when ACE node is online.)
6	If Host Name has been automatically determined, press <Tab>. Or, key in name assigned to the host pc for the ACE node. There is a 255-character limit on this field. Press <Tab> (Valid Host Name entry results in system automatically determining the Host IP Address, when ACE node is online.)	Moves cursor to ControlNet Connection field. 
7	Leave the box unchecked, if ACE is not connected to the ControlNet network. Check the box, if ACE is connected to the ControlNet network. Press <Tab>.	If the box is unchecked, cursor moves to Alarm Enabled selector, since the preceding fields are unavailable. Go to Step 9. If the box is checked, cursor moves to ACE Mac Address field.
8	Key in the Media Access Control (MAC) address assigned to the PCIC card installed the ACE node for ControlNet connections. Press <Tab>.	Moves cursor to the Alarm Enabled selector.
9	The default is alarming enabled. Remove the check to disable alarming. Press	Moves cursor to the Time Zone field.

Step	Action	Result
	<Tab>	
	TIP The time zone represents the offset value from the Greenwich Mean Time (GMT) based on your geographical location. For example, the time zone value for a CPM located in the Eastern time zone of the United States that is currently not observing daylight savings time would be -05.0 or -5. Always use the offset value that is not adjusted for daylight savings time as the entry for the Time Zone field. For example, the adjusted offset value for the Eastern time zone of the United States is -04.0, but use the unadjusted value of -5 instead. You may want to visit the http://www.worldtimeserver.com/ website, if you have a question about the appropriate offset value for your given location.	
10	Key in the appropriate time zone offset value for the location where the ACE is installed. Press <Tab>.	Moves cursor to Daylight Savings Time check box. 
11	Leave box unchecked, if Daylight Savings Time is not currently being observed at your location. Or, Check the box, if Daylight Savings time is currently being observed at your location. Press <Tab>.	Moves cursor to Year Format field. 
12	Accept default or click down-arrow button and select desired format from the list. Press <Tab>.	Moves cursor to Weekday Format field. 
13	Accept default or click down-arrow button and select desired format from the list. Click the Server History tab.	Calls up the Server History configuration form.

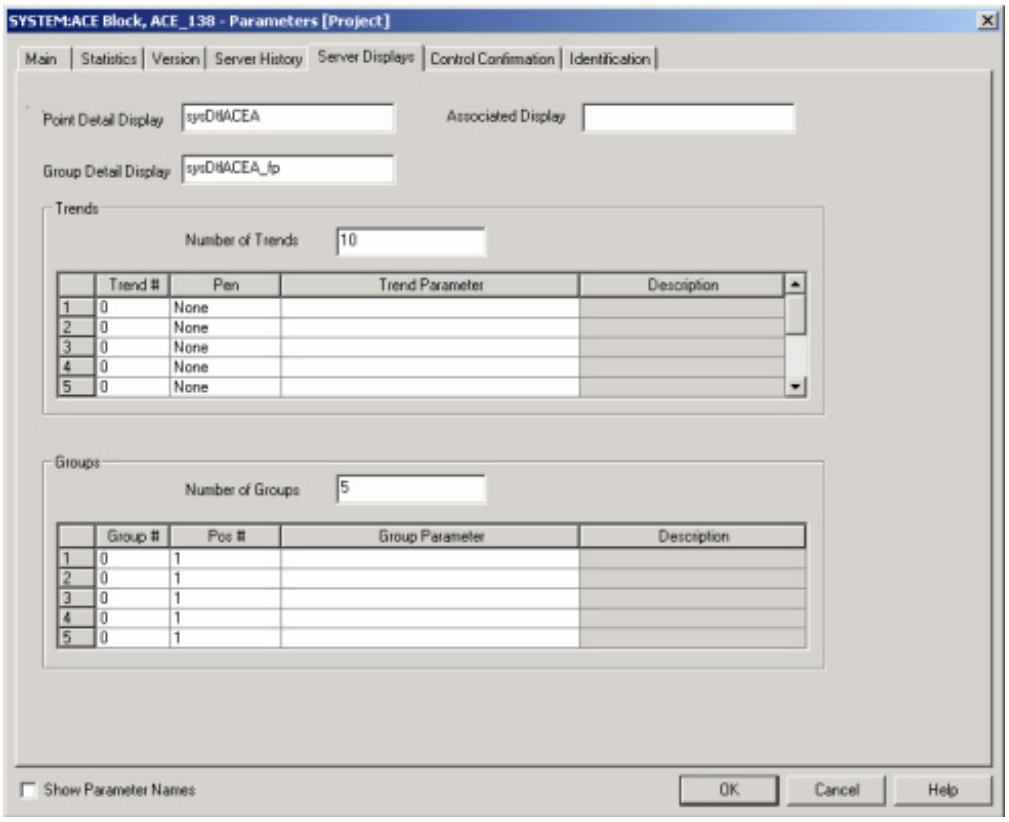
Step	Action	Result

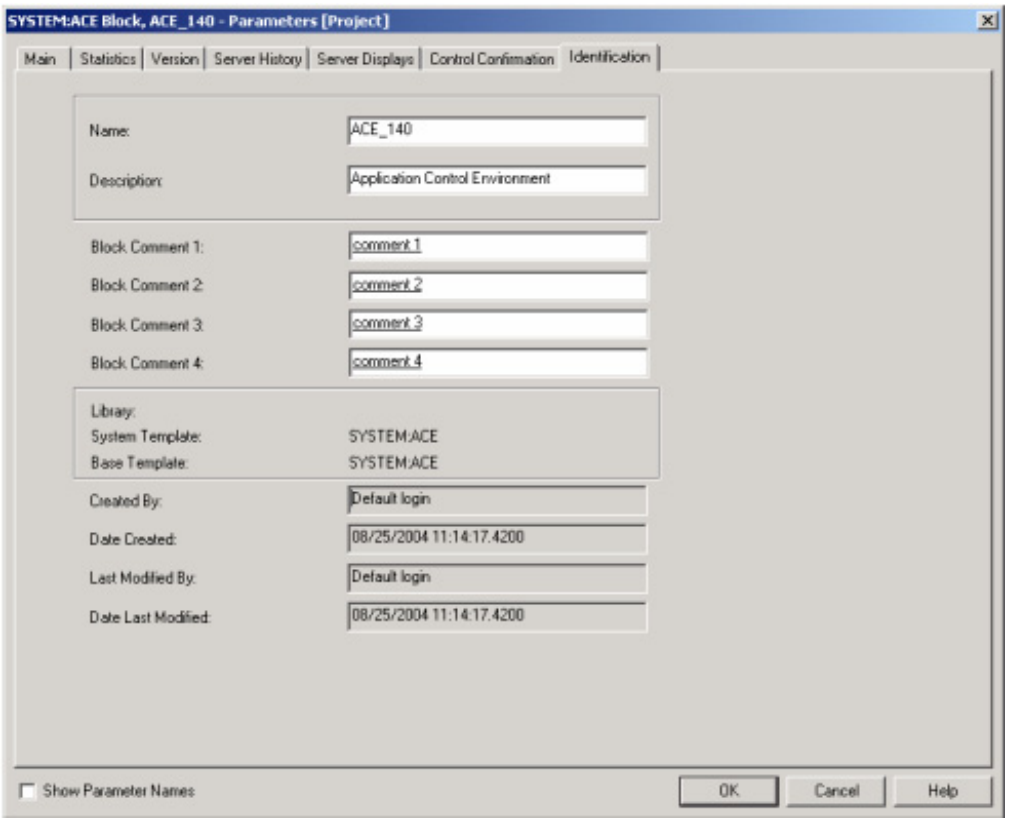
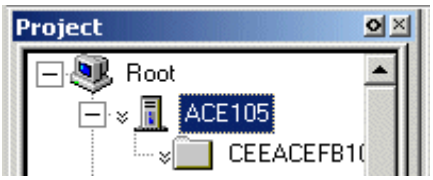


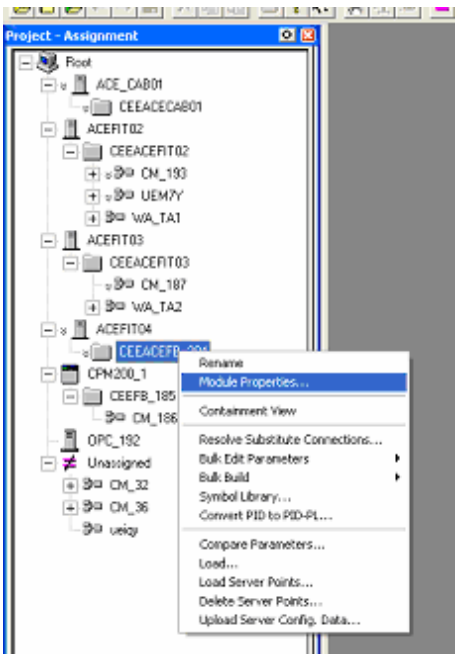
TIP

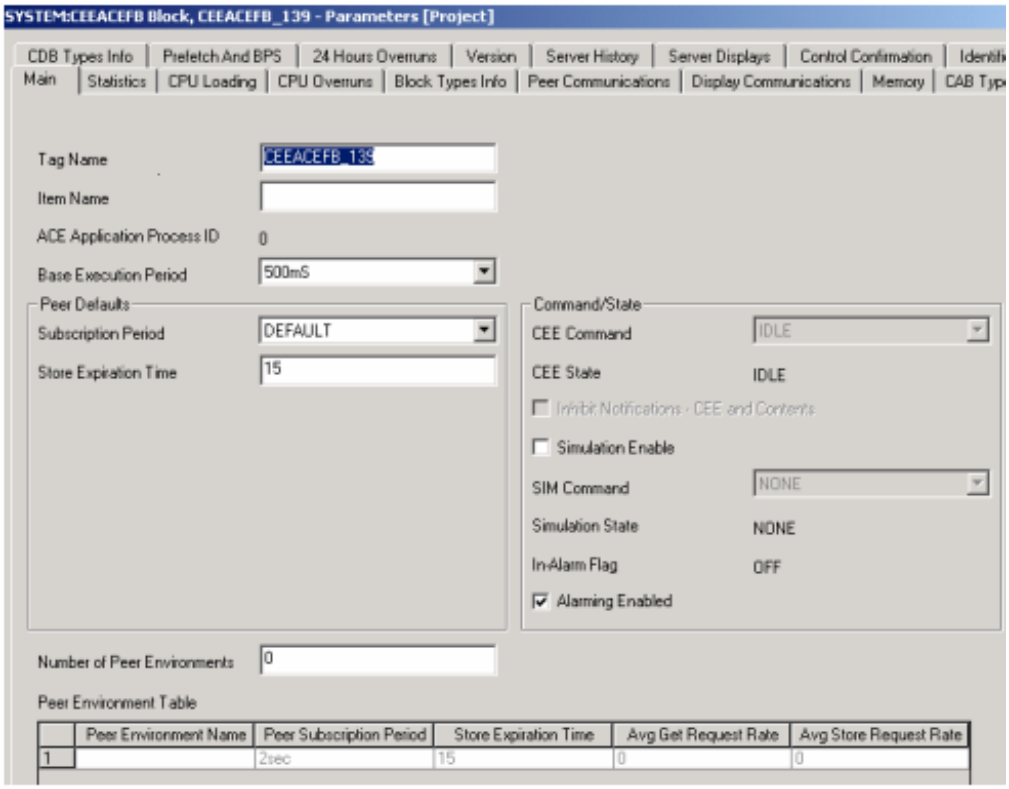
If you have a Distributed Server Architecture (DSA), you must enter the Control Area assignment for this Server. (Note that area code assignments are made through the Station application.) If you do not have a DSA, you can skip this field if areas is not enabled through the Station application.

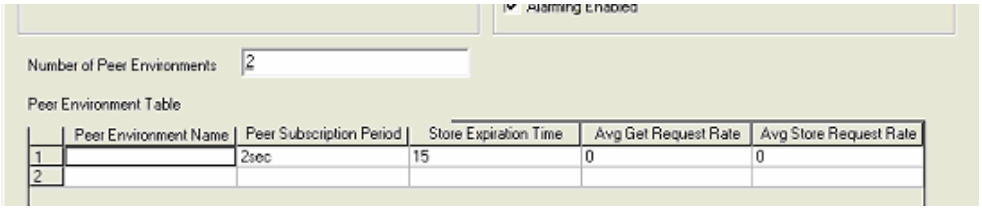
14	Use the on-line help as a guide to complete the configuration entries on this tab. Click the Server Displays tab.	Calls up the Server Displays configuration form.

Step	Action	Result
		
15	Use the on-line help as a guide to complete the configuration entries on this tab. Click Identification tab.	Calls up the Identification configuration form

Step	Action	Result
		
16	Use the on-line help as a guide to complete the configuration entries on this tab. Click the **OK** button.	Closes the form and creates ACE/CEE block icons in Project tab. 
17	Right-click CEEACE block icon.	Calls up shortcut menu.

Step	Action	Result
		
18	Click Module Properties.	Calls up CEE Block configuration form.

Step	Action	Result
		
19	Key in desired name of up to 16 characters or accept the default. Press <Tab>	Moves cursor to the Item Name field
20	If the CEE is part of an Enterprise Model, enter its Item Name here. Press <Tab>	Moves cursor to Base Execution Period field.
21	Accept the default. Press <Tab>.	Moves cursor to Subscription Period field.
22	Accept the DEFAULT or click down-arrow button and select desired period. Press <Tab>.	Moves cursor to Store Expiration Time field.
23	Accept the default or key in desired	Moves the cursor to the Simulation

Step	Action	Result
	value. Press <Tab>.	Enable selector.
24	Select if this node is to be used as a SIM ACE. (See SIM ACE User Guide for additional details). Press <Tab>.	Moves cursor to Alarming Enabled field.
25	The default is alarming enabled. Remove the check to disable alarming. If peers are to be configured, move to the *Number of Peer Environments* field and go to item 25. Otherwise, click on the Prefetch and BPS tab and go to Step 30	
Peer Configuration		
		
	TIP The Number of Peer Environments and Peer Environment Table are interactive. The value entered for the Number of Peer Environments determines how many rows appear in the Peer Environment Table.	
26	Key in number of peer environments for this ACE. Press <Tab>. Or, skip this field, if no peer environments will be used, click the Prefetch and BPS tab and go to Step 30	If peer environments will be used, cursor moves to the Peer Environment Name column in the Peer Environment Table.
27	Key in valid name for existing peer environment. Press <Tab>.	Moves cursor to Peer Subscription Period column.
28	Accept default or click down-arrow button to select another value specific to the given environment. Press <Tab>.	Moves cursor to Store Expiration Time column.
29	Accept the default value of 15 or enter a non-zero value. Return to Step 27 to enter data for another peer, or click the	

Step	Action	Result
	Prefetch and BPS tab and go to step 30	
Prefetch and BPS Configuration		
30	The only item to be configured is the Hold Breath Trippoint, used when peers fail to respond. Enter the desired value or leave at 0. Click the Server History tab	Calls up the Server History configuration form
31	Use the on-line help as a guide to complete the configuration entries on this tab. Click the Server Displays tab.	Calls up the Server Displays configuration form.
32	Use the on-line help as a guide to complete the configuration entries on this tab. Click the OK button.	Completes CEE configuration and closes the form.
33	This completes the ACE/CEE creation procedure.	

Creating an External Server

Use the following procedure to create an OLE for Process Control (OPC) Server in the Project Tree to represent an external server.

Prerequisites:

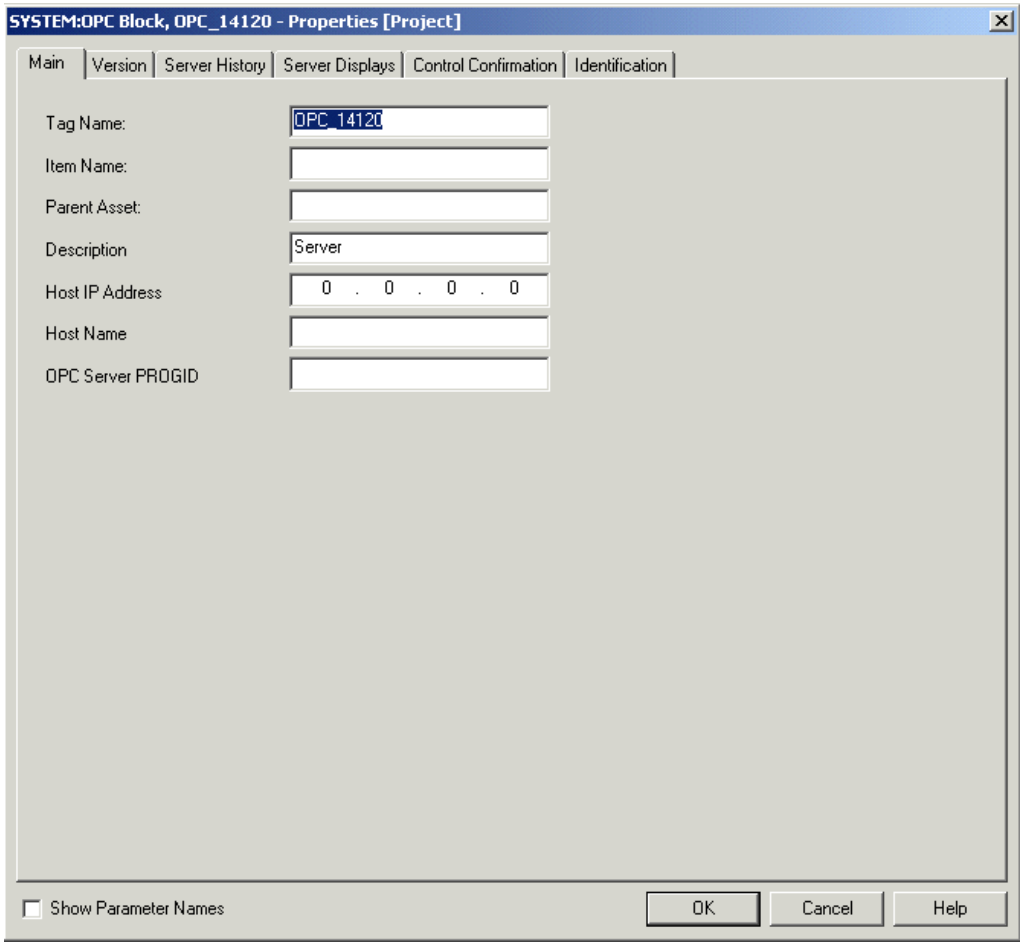
- Control Builder is running
- Tree windows are open

Considerations:

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

Step	Action	Result
1	Click File -> New -> External Servers -> OPC Server.	Calls up the OPC Server configuration form with Tag Name field highlighted.

Step	Action	Result



TIP

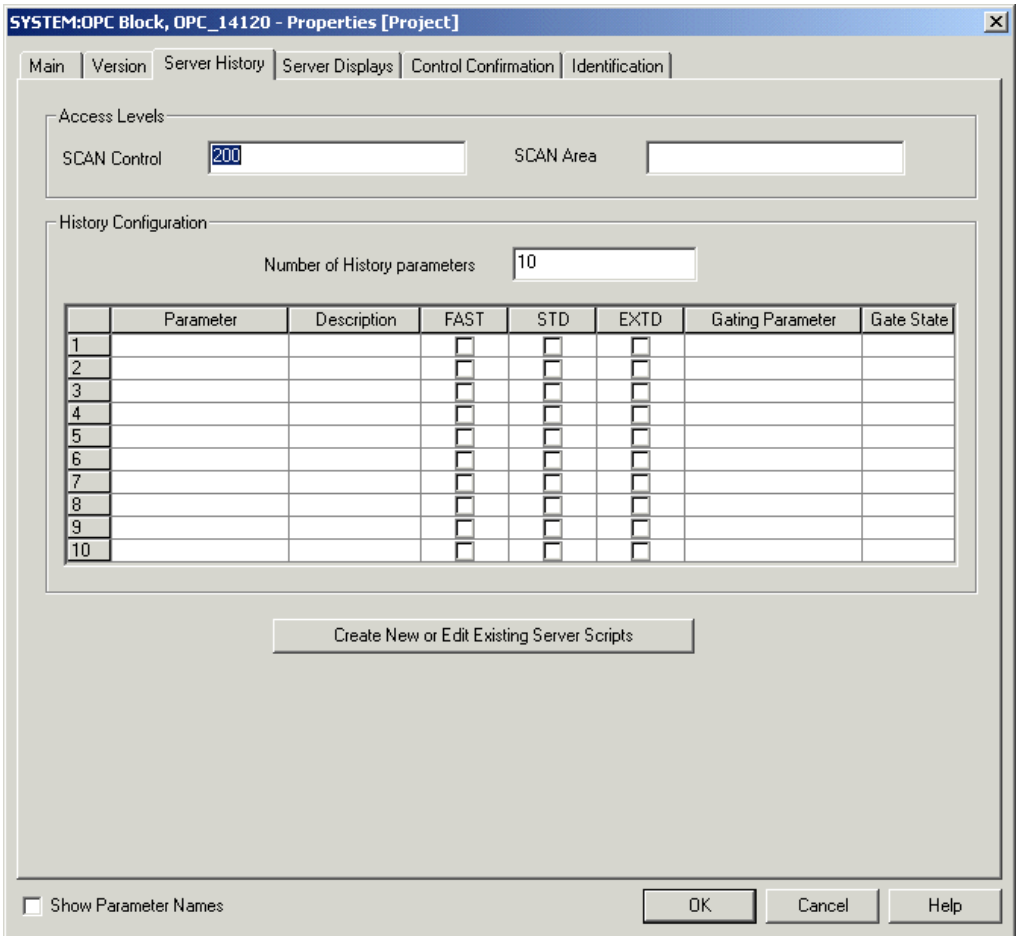
Host name and Host IP Address are interactive entries. We recommend that you key in the Host IP Address first and let the system determine the Host Name automatically. This is especially true if you are configuring the OPC Server block with the server node offline. In this case, entering the Host IP address first generates a Warning message, but entering the Host Name first generates an error message.

When keying in an IP address, use the mouse or the left and right arrow keys to move

Step	Action	Result
	the cursor to locations within the field. Do not press the <Tab> key until the complete address is keyed in. See steps 6 and 7 below.	
2	Key in desired name of up to 16 characters or accept the default. Press <Tab>.	Moves cursor to the Item Name field.
3	Enter Item Name. Press <Tab>.	Moves cursor to the Parent Asset field.
4	Enter Parent Asset name. Press <Tab>.	Moves cursor to Description field. 
5	Accept default entry or key in desired descriptive data. There is a 132-character limit on this field. Press <Tab>.	Moves cursor to Host IP Address field. 
6	Key in the host pc IP address for the OPC server node. Press <Tab>. Or, press <Tab> to skip this field and enter Host Name instead. Acknowledge any error message prompts.	System automatically determines the Host Name, when OPC server node is online, and moves cursor to Host Name field.  (Valid IP address entry results in system automatically determining the Host Name, when OPC server node is online.)
7	If Host Name has been automatically determined, press <Tab>. Or, Key in name assigned to the host pc for the OPC server node. There is a 255-character limit on this field. Press <Tab> (Valid Host Name entry results in system automatically determining the Host IP Address, when OPC server node is online.)	Moves cursor to OPC Server PROGID field. 

Step	Action	Result
	**TIP** Base the choice of version specific or general type PROGID on how "automatic" the OPC Server component version upgrade is to be. With the general type of PROGID, the OPC Server component version upgrade will be more automatic, since it requires less user interaction.	
8	Key in the Microsoft COM object Program ID for the intended OPC Server. Click the Server History tab.	Calls up the Server History configuration form.

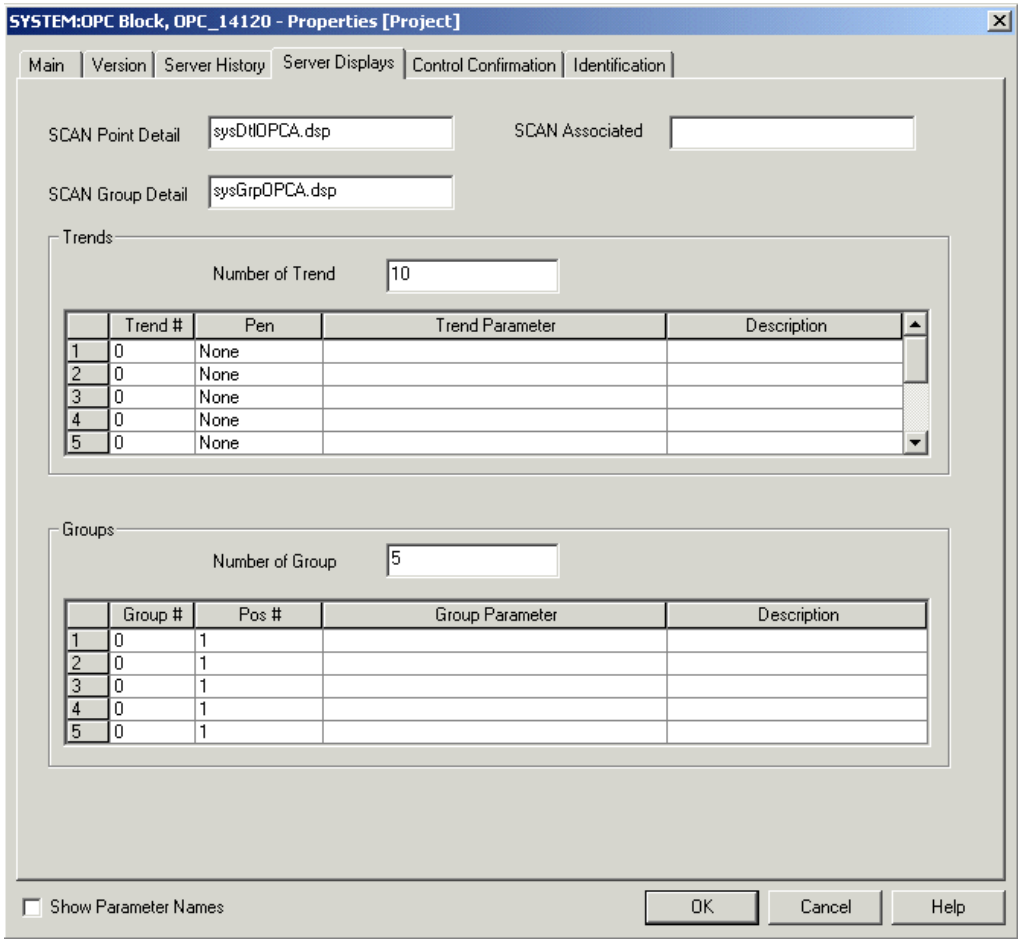
Step	Action	Result



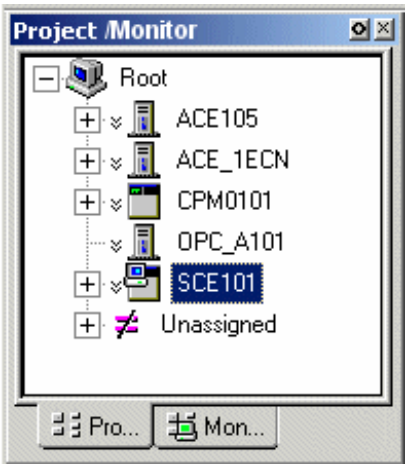
TIP

If you have a Distributed Server Architecture (DSA), you must enter the Control Area assignment for this Server. (Note that area code assignments are made through the Station application.) If you do not have a DSA, you can skip this field if areas is not enabled through the Station application.

Step	Action	Result
9	Use the on-line help as a guide to complete the configuration entries on this tab. Click the Server Displays tab.	Calls up the Server Displays configuration form.



10	Use the on-line help as a guide to complete the configuration entries on this tab. Click the OK button.	Closes the form and creates OPC Server block icon in Project tab.

Step	Action	Result
		
11	This completes the OPC Server creation procedure.	

Creating an Inter Cluster Gateway Block

Use the following procedures to create an Inter Cluster Gateway Block that will allow data transfer between Experion node clusters. For details, see the Experion Cluster Communication for ACE Interface Reference.

Prerequistes:

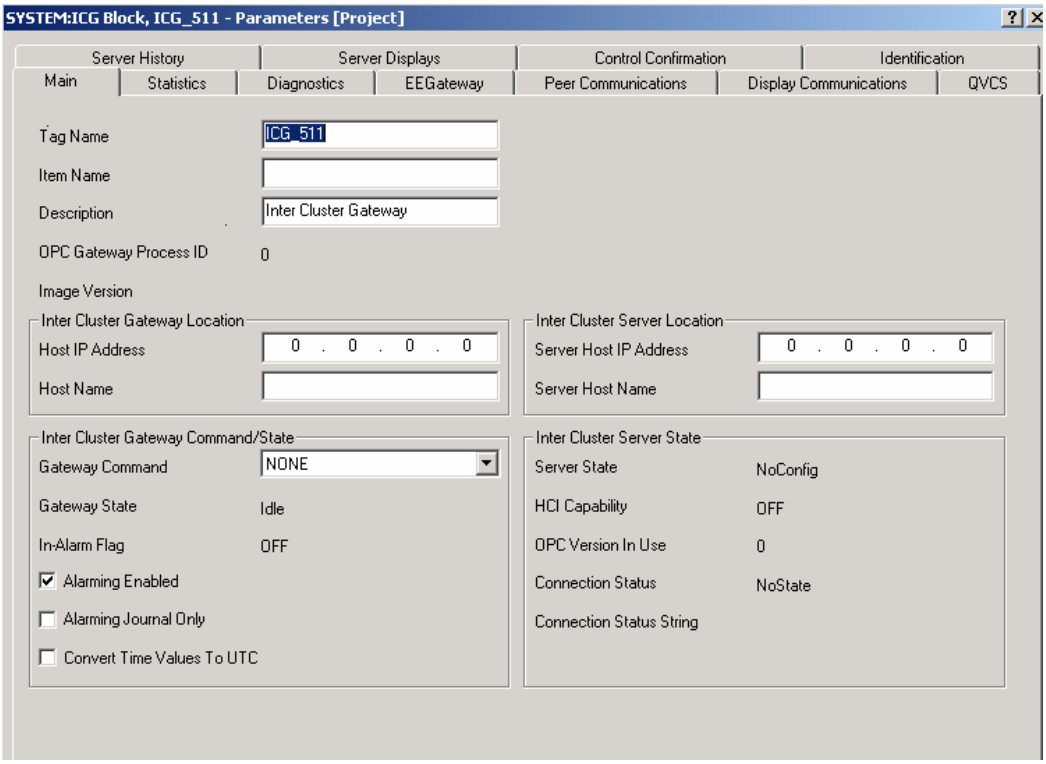
- Control Builder is running
- Tree windows are open
- You have configured assets through Enterprise Model Builder

Considerations:

- All illustrations used in the procedure are for example purposes only.
- You know what the Internet Protocol (IP) addresses are for the computers hosting the Inter Cluster Gateway and the I-C Server.
- You can use the names for the computers hosting the Inter Cluster Gateway and the I-C Server and allow the system to resolve the names into IP addresses.

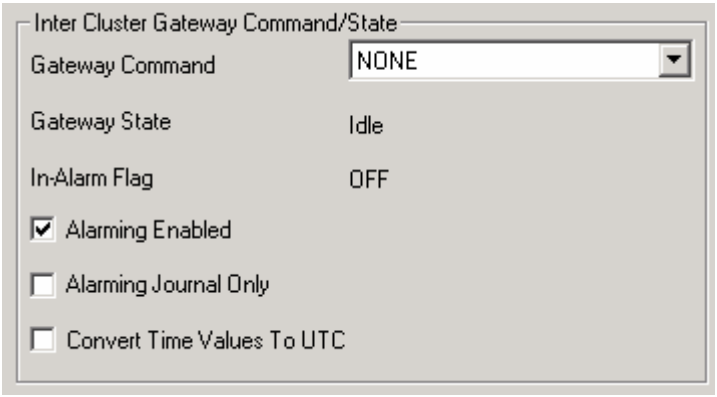
However, using the IP address results in a more robust configuration and minimizes the possibility of error conditions occurring

Step	Action	Result
1	On the **File** menu, click **New>External Servers>ICG - Inter Cluster Gateway**	Calls up the Inter Cluster Gateway (ICG) configuration form with the **Tag Name** field highlighted



2	In the **Tag Name** field, type a unique name for the block or accept the default name. Press the **Tab** key.	Moves cursor to the **Item Name** field
3	In the **Item Name** field, type the name of the Entity that this node will be associated with in the Enterprise Model Builder hierarchy. Press the **Tab** key.	Moves the cursor to the **Inter Cluster Gateway Location > Host IP Address** field

Step	Action	Result
		
4	In the **Inter Cluster Gateway Location > Host IP Address** field, type the IP address for the computer hosting the ICG function (an ACE node in the same cluster as the ICG). Press the **Tab** Key Or, press the **Tab** key to skip this field and enter **Inter Cluster Gateway Location > Host Name** instead. Acknowledge any error message prompts.	Moves the cursor to the **Host Name** field 
5	In the Inter Cluster Gateway Location **Host Name** field, check that the name of the computer is correct for the entered IP Address. The name should be entered automatically, if the Server is connected to the ACE node containing the ICG. If not, type the name of the host computer. Press the **Tab** key	Moves the cursor to the **Inter Cluster Location > Server Host IP Address** field 
6	In the **Inter Cluster Location > Server Host IP Address** field, type the IP address for the computer hosting the I-C Server that is communicating through this ICG. Press the **Tab** Key	Moves the cursor to the **Inter Cluster Location > Server Host Name field** 
7	In the **Inter Cluster Location > Server Host Name** field, check that the name of the computer is correct for the entered IP Address. The name should be entered automatically, if the Server is connected to the ACE node containing the OPC Gateway that is connected to the OPC server. If not, type the name of the host computer. Press the **Tab** key.	Moves the cursor to the **Inter Cluster Gateway Command/State > Gateway Command** field.
	TIP The PROGID field is not configurable for the ICG; when using an OPC Gateway Block to connect to an ICG, use “Hci.OPCGateway”	

Step	Action	Result
		
8	Use the default **Gateway Command** state. Press the **Tab** key.	Moves the cursor to the **Inter Cluster Gateway Command/State > Alarming Enabled** field
9	Click the **Alarming Enabled** check box to select (enable) the option. Or, clear the check box to disable the option. Press the **Tab** key.	Moves the cursor to the **Inter Cluster Gateway Command/State > Alarming Journal Only** field
10	Click the **Alarming Journal Only** check box to select (enable) the option. Or, clear the check box to disable the option. Press the **Tab** key.	Moves the cursor to the **Inter Cluster Gateway Command/State > Convert Time Values** field
11	Click the **Convert Time Values** to UTC check box to select (enable) the option. Or, clear the check box to disable the option.	Completes Main Tab data entry.
	Click the remaining tabs in succession and click the **Help** button for more information about a parameter entry field on a given tab.	
	Click the **OK** button to close the **Inter Cluster Gateway Block Parameters** form and save your configuration entries.	This completes the setup of the Inter Cluster Gateway

Ancillary activity: REEOUT block and SP push

Creation of an Inter Cluster Gateway is associated with control schemes that extend over two Experion clusters. Other activities/requirements associated with inter-cluster cascades are

- Creation of Remote EEOUT (REEOUT) blocks for the transfer of SP data from the primary Experion cluster to regulatory points in the secondary Experion cluster. See the sections on the REEOUT block in *Control Builder Components Theory* and *Control Builder Components Reference.*
- Ensuring that target points in the secondary Experion cluster are capable of accepting SP values from the ICG in their cluster. This requires that the target points have their PUSHSP parameter activated.

Creating an IOLIM and IOLINK

Use the following procedure to create an I/O Link Interface Module (IOLIM) and associated I/O Link blocks in the Project Tree that will represent an installed IOLIM.



TIP

You can configure an IOLIM block in the Control Builder Project tab without the IOLIM hardware being installed. However, it is good idea to have the communications driver and hardware that is going to be used for the system installed, configured, and running. The IOLIM needs the name of the communications driver specified on its configuration form to complete its configuration data. Like the CPM, the IOLIM represents a hardware module and the block configuration specifies the communication path to the hardware.



TIP

For Analog Output (AO) and Digital Output (DO) connections, CMs executing at 50 mS are not supported.

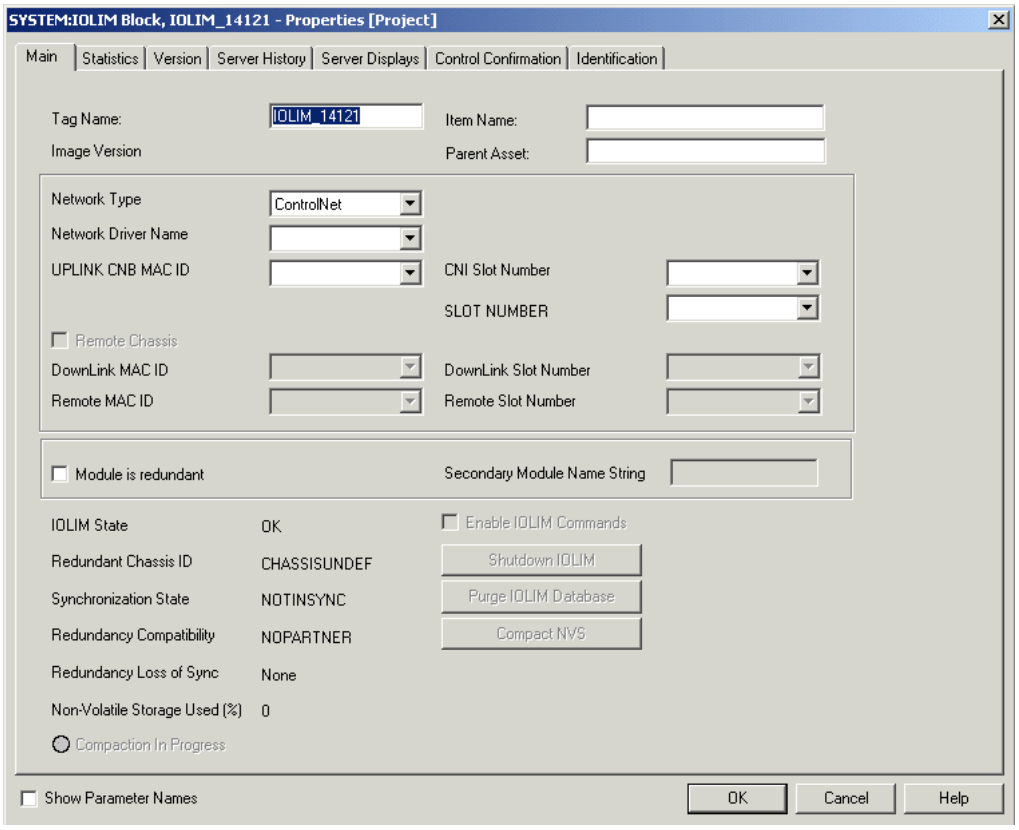
Prerequisites:

- Control Builder is running
- Tree windows are open

Considerations:

- The IOLIM provides interface to Process Manager I/O through redundant I/O links to an Input/Output Processor (IOP) card file.
- The IOLIM also supports Controller redundancy. In this case, matching IOLIMs are installed in the Redundant Chassis Pair (RCP).
- All illustrations used in the procedure are for example purposes only.

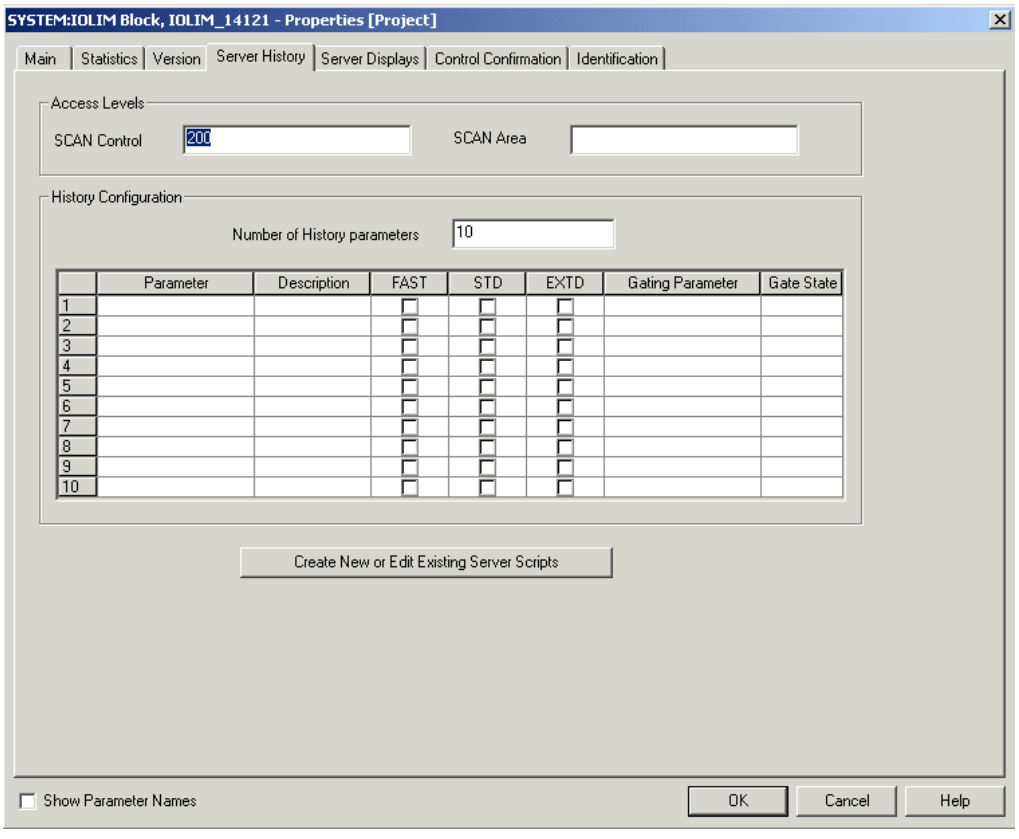
Step	Action	Result
1	Click File -> New -> Interface Modules->IOLIM - IO Link Interface Module.	Calls up the IOLIM Block configuration form with Tag Name field highlighted.



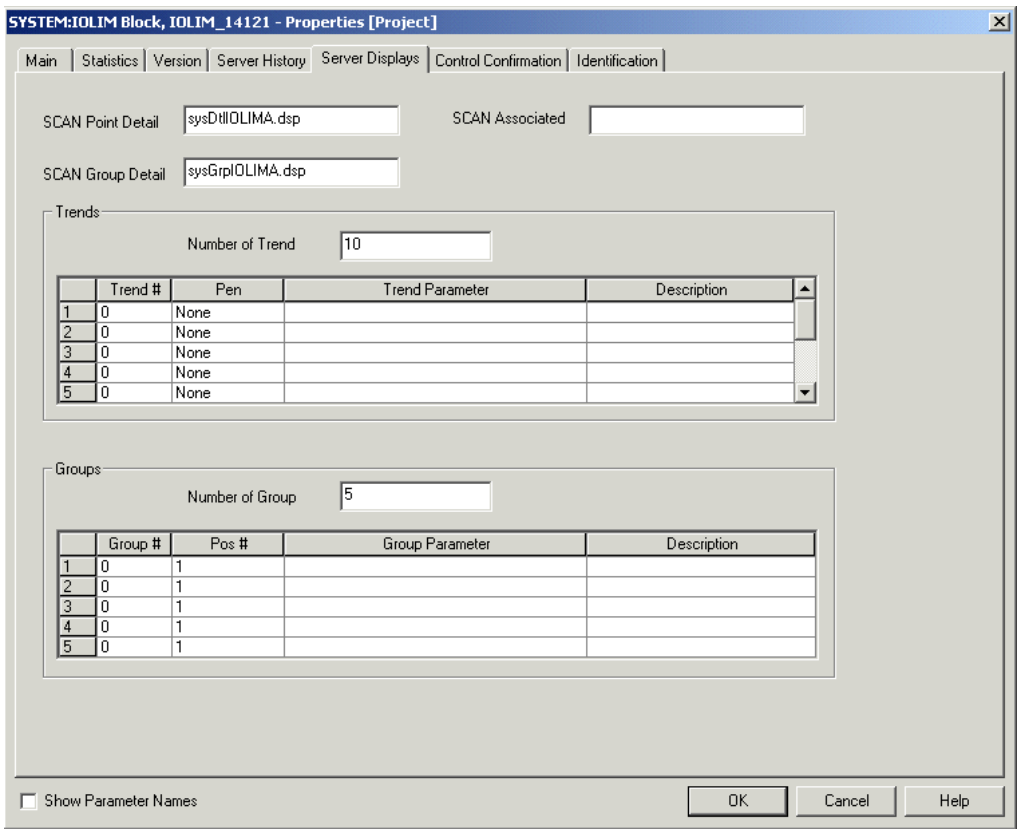
2	Key in desired name of up to 16 characters or accept the default. Press	Moves cursor to Network Type field.

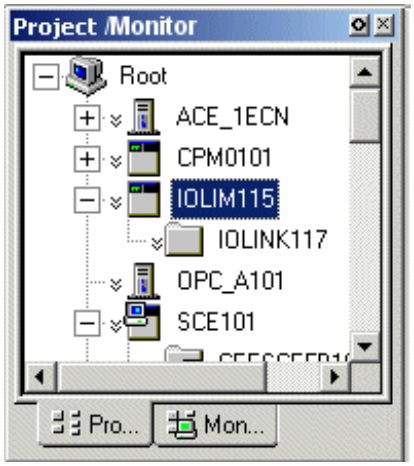
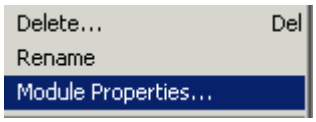
Step	Action	Result
	<Tab>	
3	Accept default or click down-arrow button and select from list. Press <Tab>.	Moves cursor to Network Driver Name field. 
	TIP All communications drivers must be installed and running before they will appear in the Driver Name selection list. The IOLIM requires a Supervisory network with ControlNet media for communications with the Server.	
4	Click down-arrow button and select installed ControlNet driver from the list. Press <Tab>.	Moves cursor to UPLINK CNB MAC ID field.
5	Click down-arrow button and select Address of the CNI (ControlNet Interface) card in the chassis that is connected to the communications adapter card in the Server. Press <Tab>.	Moves cursor to CNI Slot Number field.
6	Click down-arrow button and select slot number where supervisory CNI card is installed in the chassis. CNI card takes up one slot in the chassis and slots are numbered from left starting at 0. Press <Tab>.	Moves cursor to SLOT NUMBER field.
	ATTENTION The Remote Chassis check box and related data fields are unavailable at this time, since communications with an IOLIM installed in a remote I/O chassis is **not** supported in this release.	
7	Click down-arrow button and select slot number where IOLIM is installed in the chassis from the list. IOLIM takes up 2 slots in the chassis and slots are numbered from left starting at 0. Only	Moves cursor to Module is redundant check box.

Step	Action	Result
	odd-numbered slot is selected. Press <Tab>.	
8	Leave the box unchecked, if non-redundant CPM configuration is intended. Check the box, if IOLIM is part of a Redundant Chassis Pair (RCP). Press <Tab>.	If the box is unchecked, click Server History tab. Go to Step 10. If the box is checked, cursor moves to Secondary Module Name String field.
9	A name consisting of that of primary IOLIM followed by “SEC” will be assigned to secondary IOLIM. Accept default name or key in desired name of up to 16 characters. Click Server History tab.	Calls up the Server History configuation form. (The other data on the Main tab is unavailable in the Project mode.)



Step	Action	Result
	TIP If you have a Distributed Server Architecture (DSA), you must enter the Control Area assignment for this Server. (Note that area code assignments are made through the Station application.) If you do not have a DSA, you can skip this field if areas is not enabled through the Station application.	
10	Use the on-line help as a guide to complete the configuration entries on this tab. Click the Server Displays tab.	Calls up the Server Displays configuration form.



Step	Action	Result
11	Use the on-line help as a guide to complete the configuration entries on this tab. Click the OK button.	Closes the form and creates IOLIM/IOLINK block icons in Project tab. 
12	Right-click IOLINK block icon.	Calls up shortcut menu. 
13	Click Module Properties...	Calls up IOLINK Block configuration form with Tag Name field highlighted.
14	Key in desired name of up to 16 characters or accept the default. Press <Tab>	Moves cursor to Description field. 
15	Key in desired descriptive text of up to 132 characters. Click the OK button.	Closes the form, since no other data on this form is available in the Project mode.
16	This completes the IOLIM/IOLINK creation procedure.	

Creating Redundancy Modules

Use the following procedure to create primary and secondary Redundancy Modules (RMs).

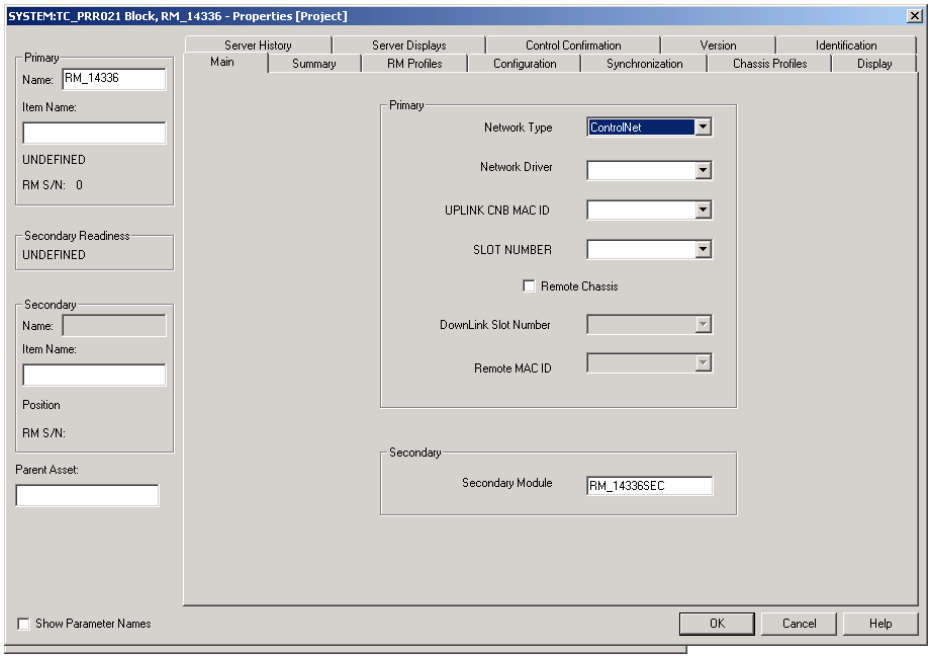
Prerequisites:

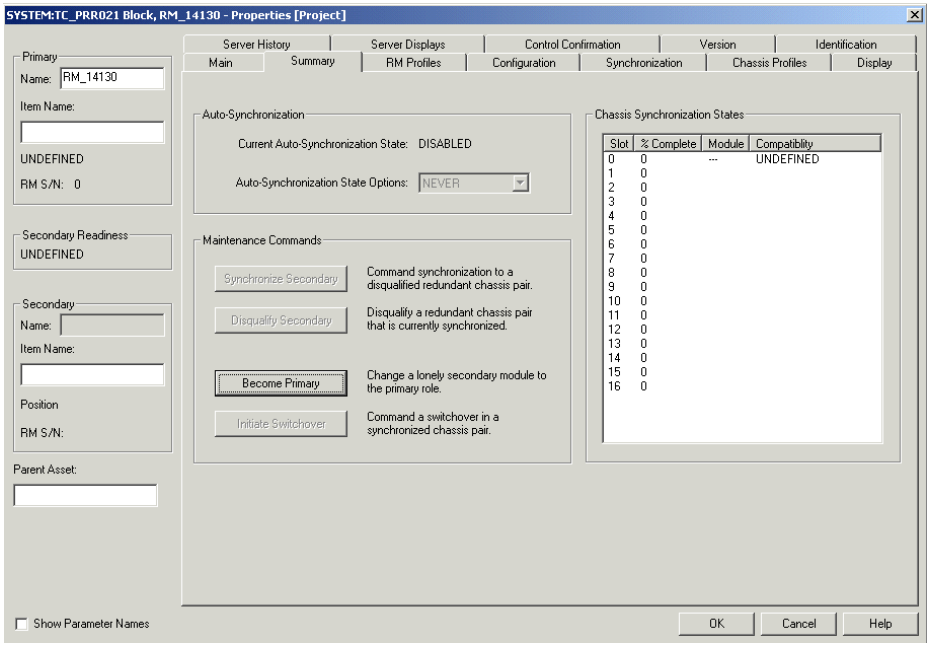
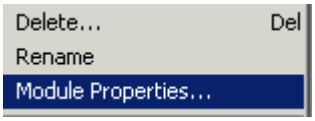
- Control Builder is running
- Tree windows are open

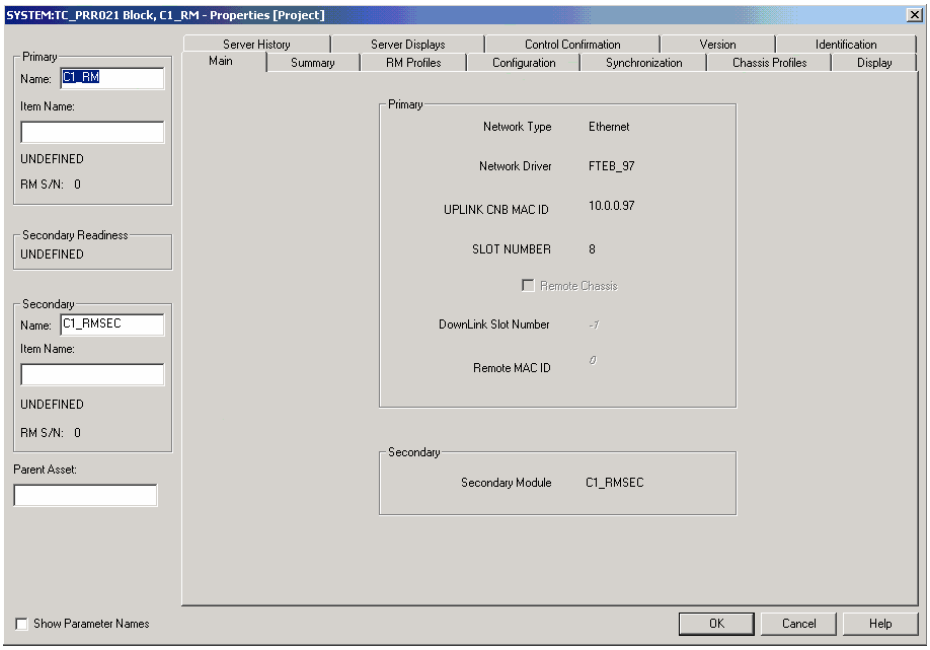
Considerations:

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

Step	Action
1	Click File -> New -> Redundancy Module.
2	By default, Redundancy Modules are created in pairs. Enter the following information to configure a Redundancy Module: • Primary: Name: Accept default or key in your own consisting of up to 16 characters. • Network Type: ControlNet or Ethernet • Network Driver: Select from a choice of currently installed and configured RSLinx drivers • UPLINK CNB MAC ID: Address of the CNI (ControlNet Interface) card in the chassis that is connected to the communications adapter card in the server. • SLOT NUMBER: RM takes up 2 slots in the chassis and slots are numbered from left starting at 0. Only odd-numbered slot is selected.
	**ATTENTION** For Secondary Module name, a name consisting of the name of the primary RM followed by “SEC” will be assigned to secondary RM.

Step	Action
	
3	Click OK.
4	Right-click the primary RM block icon and select Configure Module Parameters from the shortcut menu.
5	Using on-line help as a guide, select available configuration parameters for the primary RM for all appropriate tabs (that is, Configuration, Synchronization, Server History, Server Displays, etc.).

Step	Action
	
6	Click OK.
7	Right-click the secondary RM block icon and select Module Properties from the shortcut menu. The Block parameter's Configuration form appears on the screen. 

Step	Action
	
8	Using on-line help as a guide, select configuration parameters for the secondary RM for all appropriate tabs (Configuration, Synchronization, Server History, Server Displays, etc.).
9	Click OK.
10	This completes the RM creation procedure.

Creating an instance of Input/Output Modules

The Controller supports a variety of Input/Output data interfaces including Chassis I/O - Series A, Rail I/O - Series A, Rail I/O - Series H, and Process Manager I/O as well as links to H1 fieldbus devices and other third party networks. The following procedure is based on creating a Chassis I/O module, but it can be readily adapted to any I/O type that is included in a Control Builder library. The procedure is divided into two sub-procedures to cover the menu and drag and drop methods, respectively.

Prerequisites:

- Control Builder is running
- Tree windows are open

Considerations:

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



TIP

You must enter the address information for the I/O Module before Control Builder will create a block instance of the module in the Project tree.

Menu method

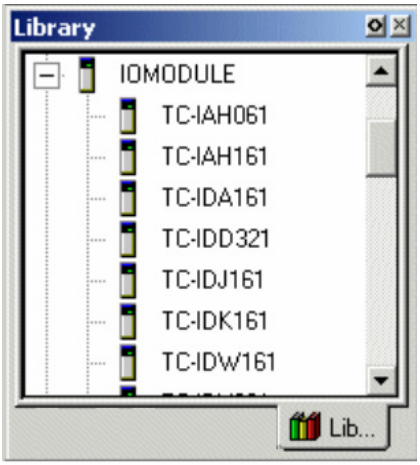
Use the following procedure to create a Chassis I/O module using the menu method.

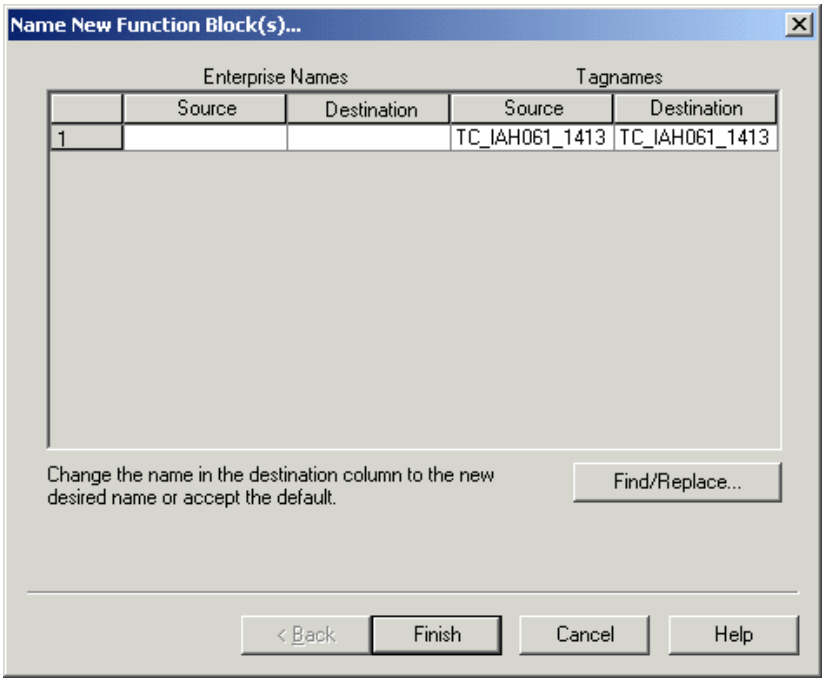
Step	Action
1	Click File -> New -> I/O Modules -> IOMODULE -> [desired I/O Module] to initiate creation of an IOM block by opening its configuration form.
2	On the Main Tab of the form, enter the appropriate I/O hardware location data in the applicable fields such as; the IOM Slotnum in Chassis, Uplink CNB in IO Rack, and Downlink CNB Slot Number for Chassis I/O. If necessary, access the on-line help for assistance during this step.
3	Click the OK button to create an instance of the I/O Module block in the Project. tree.
4	If multiple IOMs are required, repeat this procedure for each occurrence of the IOM, even if the type is the same as a previously created IOM.

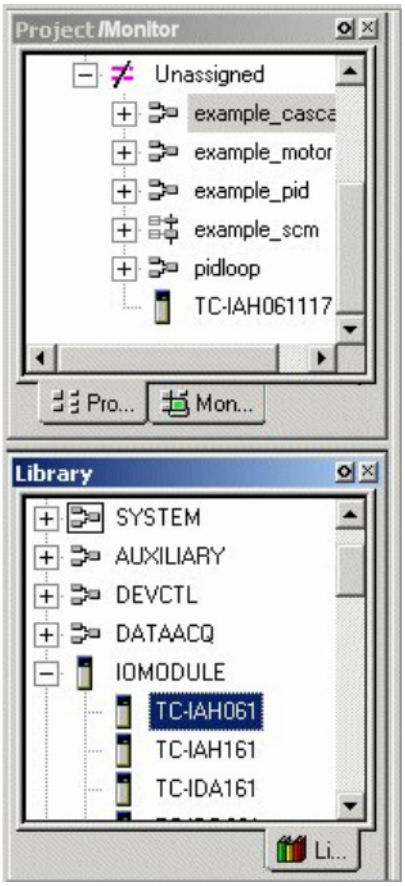
Step	Action
5	Complete the configuration of each module before loading it to the Controller. Use the appropriate configuration procedure in this document or the applicable I/O component Implementation Guide for reference along with the on-line help.

Drag and drop method

Use the following procedure to create a Chassis I/O module using the drag and drop method.

Step	Action
1	From theLibrary tree window in Control Builder, click the Library to access library contents.
2	Click the ⊞ sign for the IOMODULE icon to expand its tree. 
3	Click the Project tab in the other tree window, to access the root project contents.
4	Drag and Drop the desired I/O Module from the IOMODULE contents in the Library tree to an open area in the Project tree.
	TIP Release the mouse button when the cursor with ⊞ sign appears in Project

Step	Action
	tree to drop the module.
5	Accept the default name assigned to the module in the Name New Function Block dialog or key in a desired name of your own. Click the Finish button. 
6	Confirm that an IOM block icon now appears in the Project tree.
7	If multiple IOMs are required, repeat this procedure for each occurrence of the IOM, even if the type is the same as a previously created IOM.
8	Complete the configuration of each module before loading it to the Controller. Use the appropriate configuration procedure in this document or the applicable separate I/O component Implementation Guide for reference along with the on-line help.



Adding I/O

TIP

The number that appears after any object in Control Builder is only a reference created by the control library to ensure name uniqueness and has no other significance. In the previous example, an I/O module was created resulting in a device named TC-IAH061117. The 117 has no significance other than a reference for Control Builder.

Creating function blocks for Input/Output Modules (IOM) or Processors (IOP)

If you want to create a. . .	Then …
Series C I/O AI-HART IOM	Click here
Series C I/O AI-LLMUX IOM	Click here
Series C I/O AO-HART IOM	Click here
Series C I/O DI-24 IOM	Click here
Series C I/O DI-HV IOM	Click here
Series C I/O DO-24B IOM	Click here
PM I/O High Level Analog Input IOP	Click here
PM I/O HART High Level Analog Input IOP	Click here
PM I/O Low Level Analog Multiplexer Input IOP	Click here
PM I/O Analog Output IOP	Click here
PM I/O HART Analog Output IOP	Click here
PM I/O Digital Input IOP	Click here
PM I/O Digital Output IOP	Click here
PM I/O STIMV IOP	Click here
PM I/O High Level Analog Input IOP	Click here

Creating Series C I/O AI-HART

The following section is a general guide for creating and configuring a function block for a Series C I/O AI-HART.

Refer to the Series C I/O Guide for a more detailed explanation of configuring a Series C I/O AI-HART.

Prerequisites:

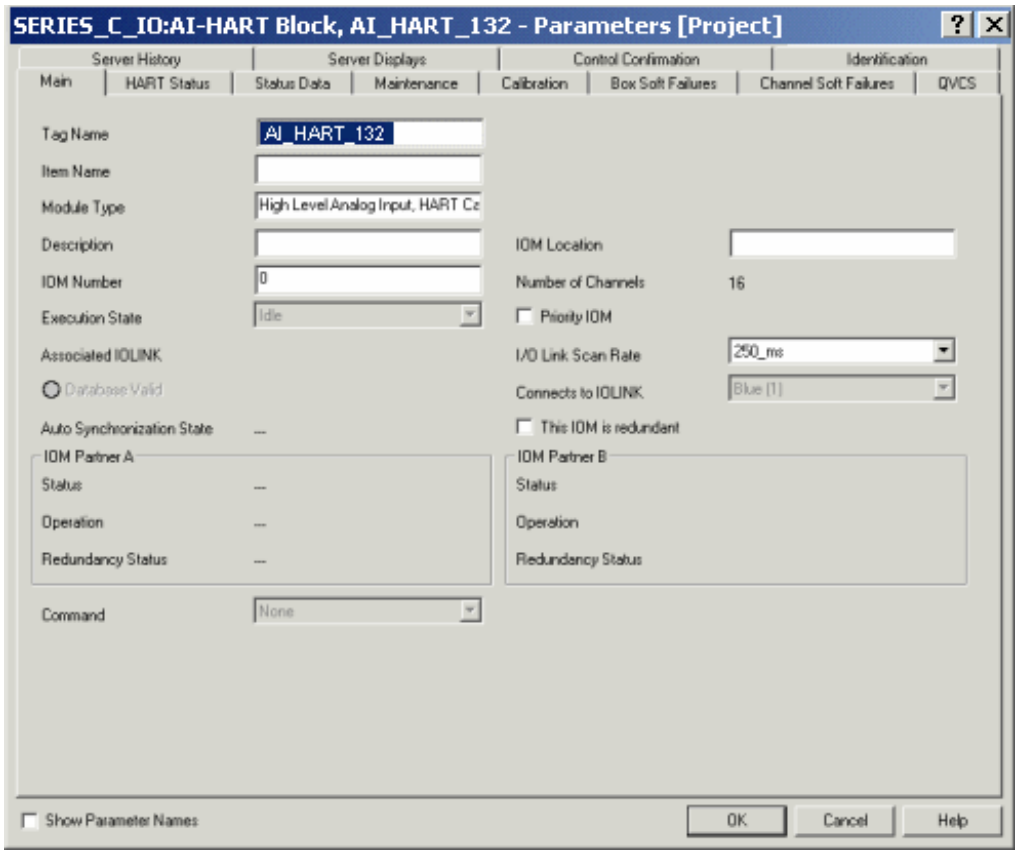
- Control Builder is running
- Tree windows are open

Considerations:

- The following procedure uses the menu method of creation but the Drag and Drop Method described above for Input/Output Modules can also be used
- All illustrations used in the procedure are for example purposes only.

Step	Action	Result
1	Click File->New->I/O Modules -> SERIES_C_IO -> AI-HART High Level Analog Input, HART Capable, 16 channels.	Calls up the AI-HART block configuration form with Tag Name field highlighted.

Step	Action	Result



To complete the configuration of the module, refer to: Series C I/O Guide.

Creating Series C I/O AI-LLMUX

The following section is a general guide for creating and configuring a function block for a Series C I/O AI-LLMUX.

Refer to the Series C I/O Guide for a more detailed explanation of configuring a Series C I/O AI-LLMUX.

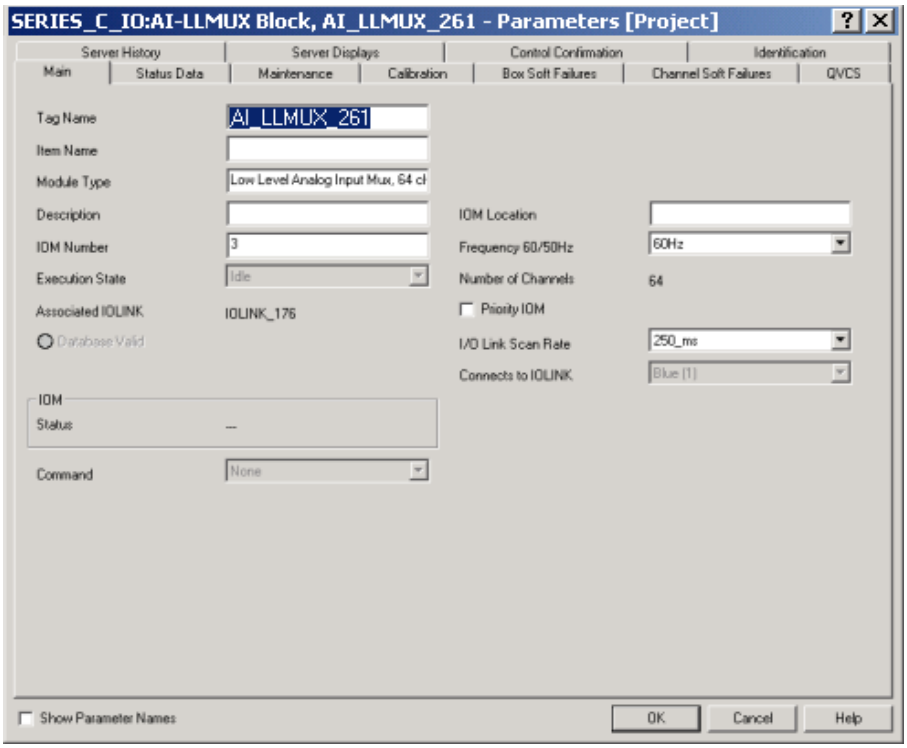
Prerequisites:

- Control Builder is running
- Tree windows are open

Considerations:

- The following procedure uses the menu method of creation but the Drag and Drop Method described above for Input/Output Modules can also be used
- All illustrations used in the procedure are for example purposes only.

Step	Action	Result
1	Click File->New->I/O Modules -> SERIES_C_I/O -> AI-LLMUX - Low Level Analog Input Mux, 64 channels.	Calls up the AI-LLMUX block configuration form with Tag Name field highlighted.



To complete the configuration of the module, refer to: Series C I/O Guide.

Creating Series C I/O AO-HART

The following section is a general guide for creating and configuring a function block for a Series C I/O AI-LLMUX.

Refer to the Series C I/O Guide for a more detailed explanation of configuring a Series C I/O AI-LLMUX.

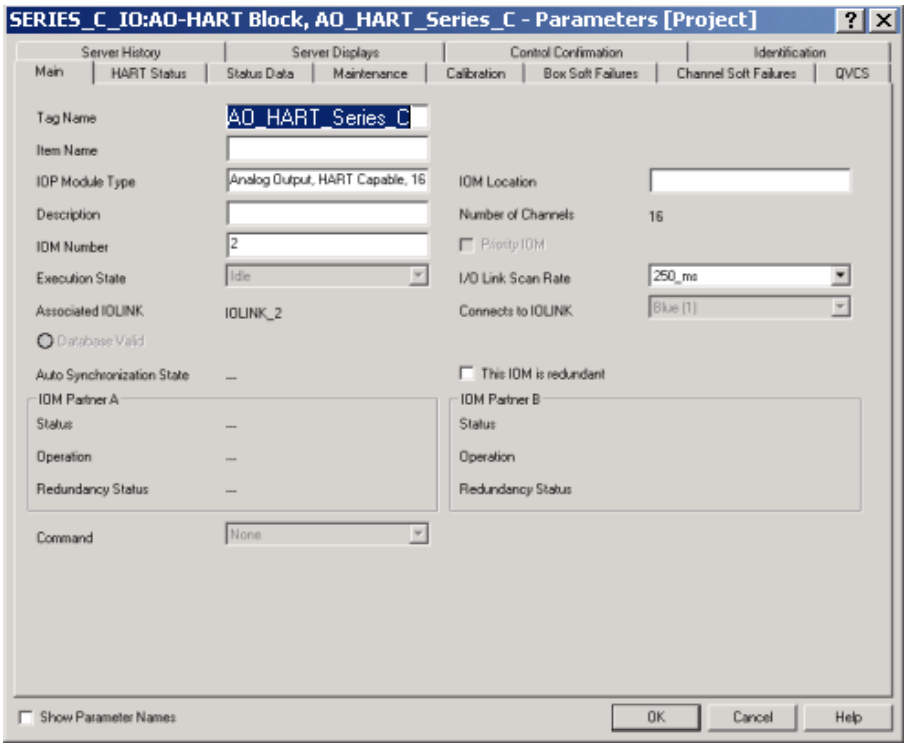
Prerequisites:

- Control Builder is running
- Tree windows are open

Considerations:

- The following procedure uses the menu method of creation but the Drag and Drop Method described above for Input/Output Modules can also be used
- All illustrations used in the procedure are for example purposes only.

Step	Action	Result
1	Click File->New->I/O Modules -> SERIES_C_IO -> AO-HART Analog Output, HART Capable, 16 channels.	Calls up the AO-HART Block configuration form with Tag Name field highlighted.

Step	Action	Result
		

To complete the configuration of the module, refer to: Series C I/O Guide.

Creating Series C I/O DI-24

The following section is a general guide for creating and configuring a function block for a Series C I/O DI-24.

Refer to the Series C I/O Guide for a more detailed explanation of configuring a Series C I/O DI-24.

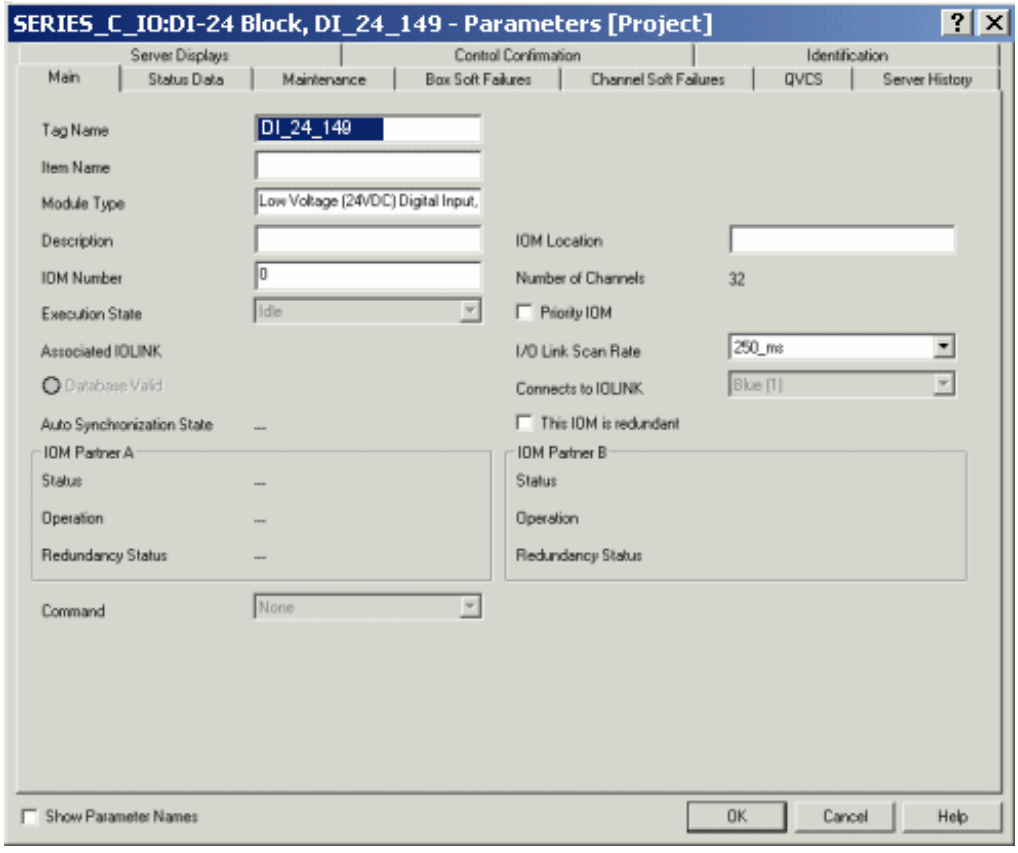
Prerequisites:

- Control Builder is running
- Tree windows are open

Considerations:

- The following procedure uses the menu method of creation but the Drag and Drop Method described above for Input/Output Modules can also be used
- All illustrations used in the procedure are for example purposes only.

Step	Action	Result
1	Click File->New->I/O Modules -> SERIES_C_IO->DI-24 - Low Voltage (24VDC) Digital Input, 32 channels.	Calls up the DI-24 block configuration form with Tag Name field highlighted.



To complete the configuration of the module, refer to: Series C I/O Guide.

Creating Series C I/O DI-HV

The following section is a general guide for creating and configuring a function block for a Series C I/O DI-HV.

Refer to the Series C I/O Guide for a more detailed explanation of configuring a Series C I/O DI-HV.

Prerequisites:

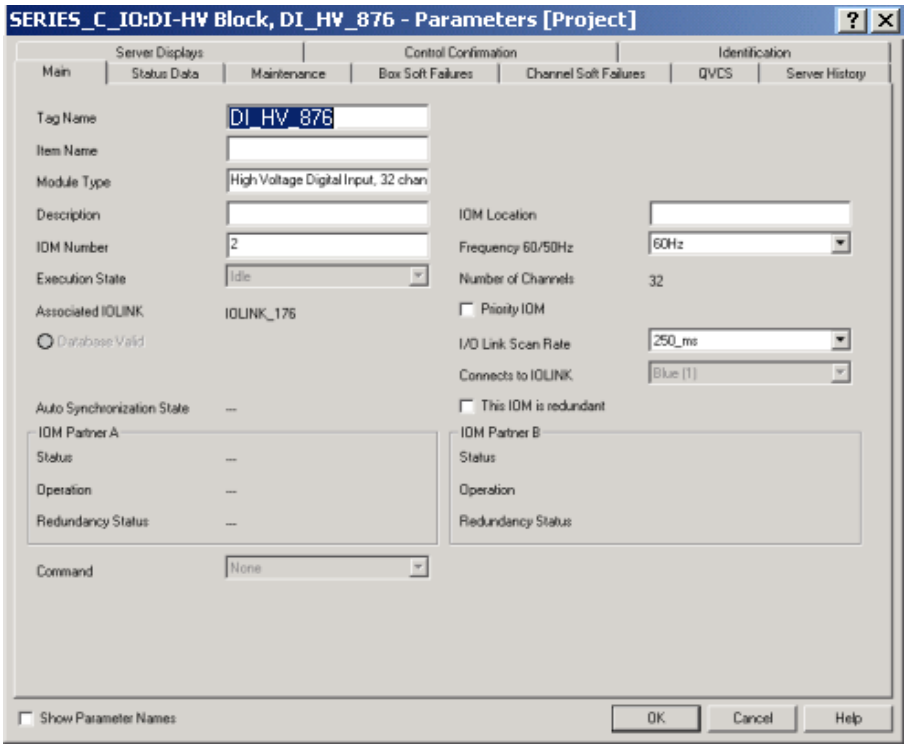
- Control Builder is running
- Tree windows are open

Considerations:

- The following procedure uses the menu method of creation but the Drag and Drop Method described above for Input/Output Modules can also be used
- All illustrations used in the procedure are for example purposes only.

Step	Action	Result
1	Click File->New->I/O Modules -> SERIES_C_IO -> DI-HV - High Voltage Digital Input, 32 channels.	Calls up the DI-HV block configuration form with Tag Name field highlighted.

Step	Action	Result



To complete the configuration of the module, refer to: Series C I/O Guide.

Creating Series C I/O DO-24B

The following section is a general guide for creating and configuring a function block for a Series C I/O DO-24B.

Refer to the Series C I/O Guide for a more detailed explanation of configuring a Series C I/O DO-24B.

Prerequisites:

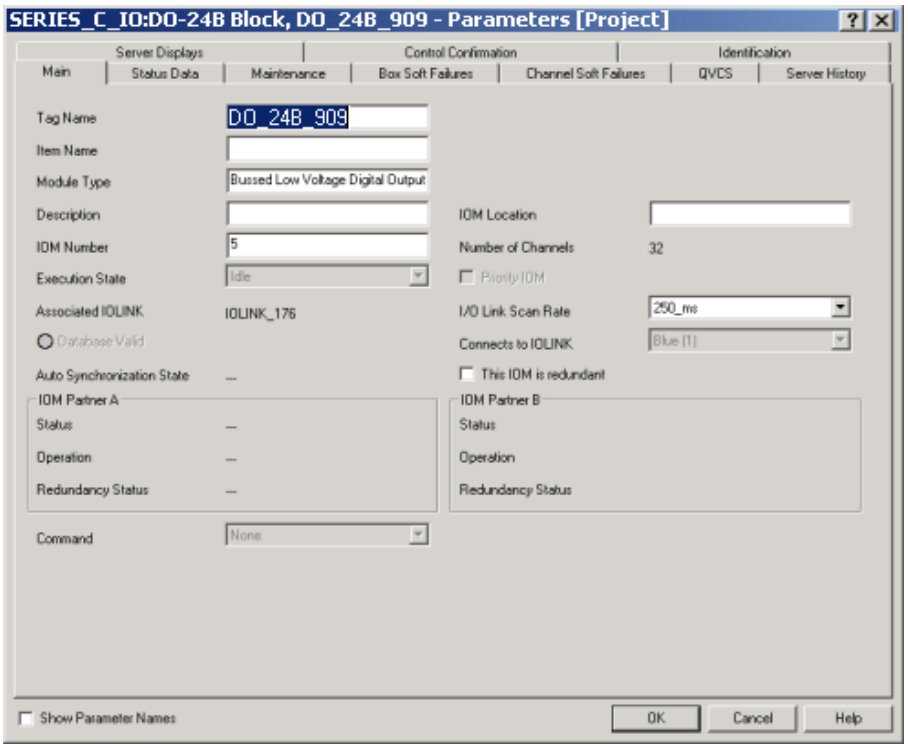
- Control Builder is running
- Tree windows are open

Considerations:

- The following procedure uses the menu method of creation but the Drag and Drop Method described above for Input/Output Modules can also be used
- All illustrations used in the procedure are for example purposes only.

Step	Action	Result
1	Click File->New->I/O Modules -> SERIES_C_IO -> DO-24B - Bussed Low Voltage Digital Output, 32 channels.	Calls up the DO-24B block configuration form with Tag Name field highlighted.

Step	Action	Result



To complete the configuration of the module, refer to: Series C I/O Guide.

Creating PM I/O High Level Analog Input IOP

Use the following procedure as a general guide for creating and configuring a function block for a High Level Analog Input (HLAI) type Input/Output Processor (IOP).

Prerequisites:

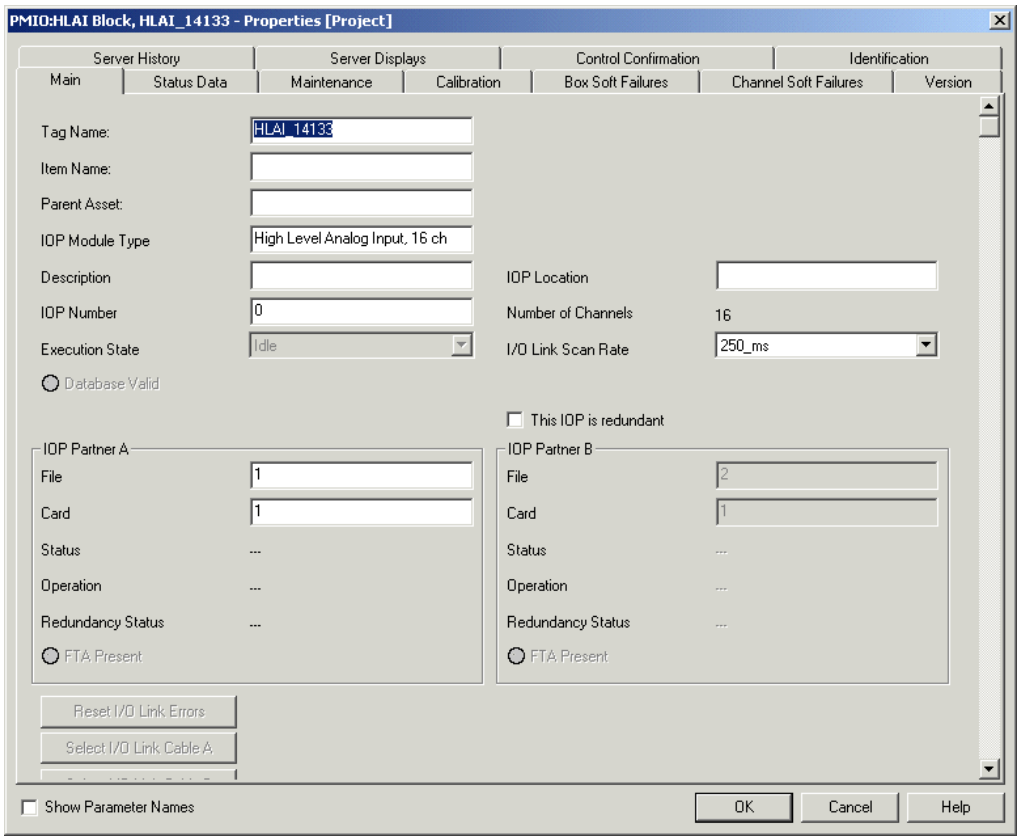
- Control Builder is running
- Tree windows are open

Considerations:

- The following procedure uses the menu method of creation but the Drag and Drop Method described above for Input/Output Modules can also be used.
- All illustrations used in the procedure are for example purposes only.

Step	Action	Result
1	Click File->New->I/O Modules->PMIO->HLAI - High Level Analog Input, 16 ch.	Calls up the HLAI Block configuration form with Tag Name field highlighted.

Step	Action	Result



To complete the configuration of the module, refer to: *Completing the configuration form*

Completing the configuration form

Prerequisites:

- Control Builder is running
- Tree windows are open

Considerations:

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

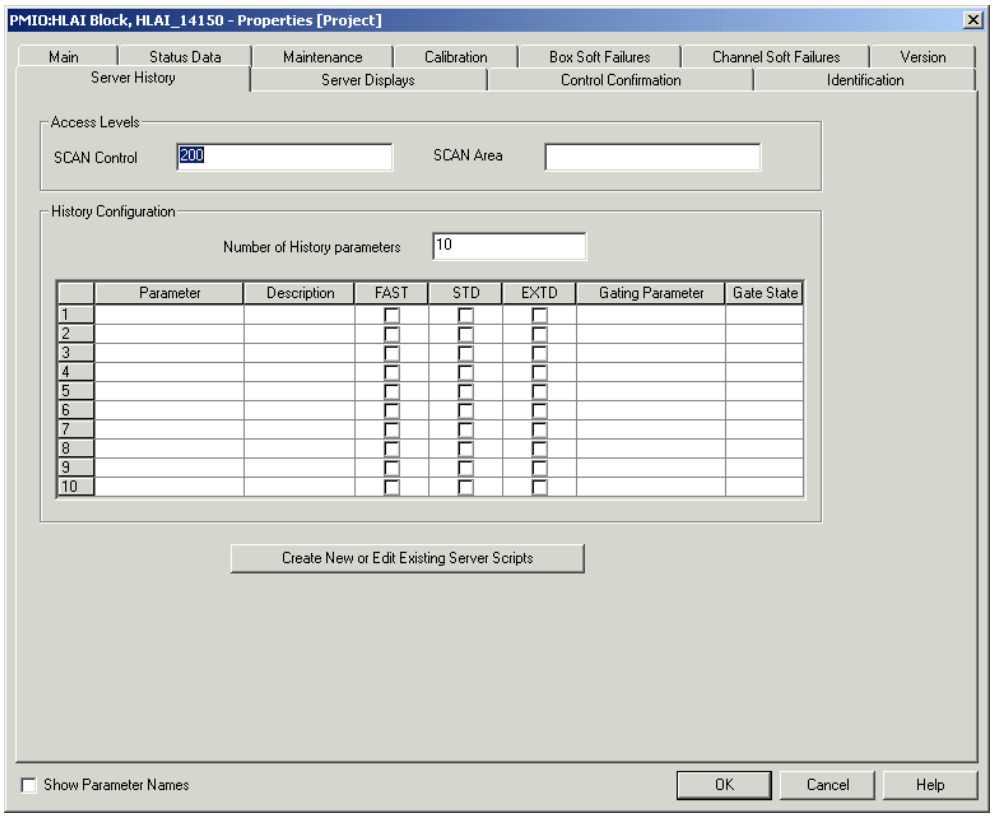
The following procedure gives the steps to complete the configuration of the configuration forms tabs.

Step	Action	Result
1	Key in desired name of up to 16 characters or accept the default. Press <Tab>.	Moves cursor to Item Name field.
2	Enter Item Name. Press <Tab>.	Moves cursor to Parent Asset field.
3	Enter Parent Asset name. Press <Tab>.	Moves cursor to IOP Module Type field. Example for PM I/O High Level Analog 
	TIP When naming a HART IOP function block, it may be helpful to use the word “HART” (or HRT) in the name to readily identify it as a HART IOP.	
4	Be sure description in field matches the desired module type. Add optional details, if desired. There is a 132-character limit on this field. Press <Tab>.	Moves cursor to Description field. 
5	Key in optional descriptive data of up to 132 characters. Press <Tab>.	Moves cursor to IOP Location.
6	Key in the logical address of the associated I/O Processor (IOP) on the given I/O Link. Press <Tab>	Moves cursor to I/O Link Scan Rate field. 
	For the following: • PM I/O High Level Analog Output IOP - HLAI • PM I/O HART Analog Output IOP HLAIHART • PM I/O Digital Input IOP DI24V	Go to Step 7

Step	Action	Result
	For the following: • PM I/O Analog Output IOP AO16 • PM I/O HART Analog Output IOP AO16HART • PM I/O Digital Output IOP DO32 Accept default or click down arrow and select desired rate from list. Press <Tab>	Moves cursor to Failure Option field. 
	For the following: • PM I/O Low Level Analog Multiplexer Input IOP LLMUX • PM I/O STIMV IOP Accept default or click down arrow and select desired rate from list. Press <Tab>	Moves cursor to Frequency 60/50Hz field.  LLMUX go to Step 8. STIMV got to Step7
7	Accept default or click down arrow and select desired rate from list. Press <Tab>	Moves cursor to This IOP is redundant check box. 
8	Leave box unchecked if IOP is nonredundant. Press <Tab>. Check box if IOP is one of a redundant pair. Press <Tab>.	Moves cursor to IOP Partner A, File field. 
9	Accept default or key in the IOP card file number for the primary IOP. Press <Tab>.	Moves cursor to IOP Partner A, Card field.  For nonredundant IOP, calls up Server

Step	Action	Result
		History tab. Go to Step 12.
10	Accept default or key in the card slot number for the primary IOP. • If IOP is nonredundant, click Server History tab. • If IOP is redundant, press <Tab>.	For redundant IOP, moves cursor to IOP Partner B, File field. 
11	Accept default or key in the IOP card file number for the secondary IOP. Press <Tab>.	Moves cursor to IOP Partner B, Card field. 
12	Accept default or key in the card slot number for the secondary IOP. Click Server History tab.	Calls up the Server History configuration form.

Step	Action	Result

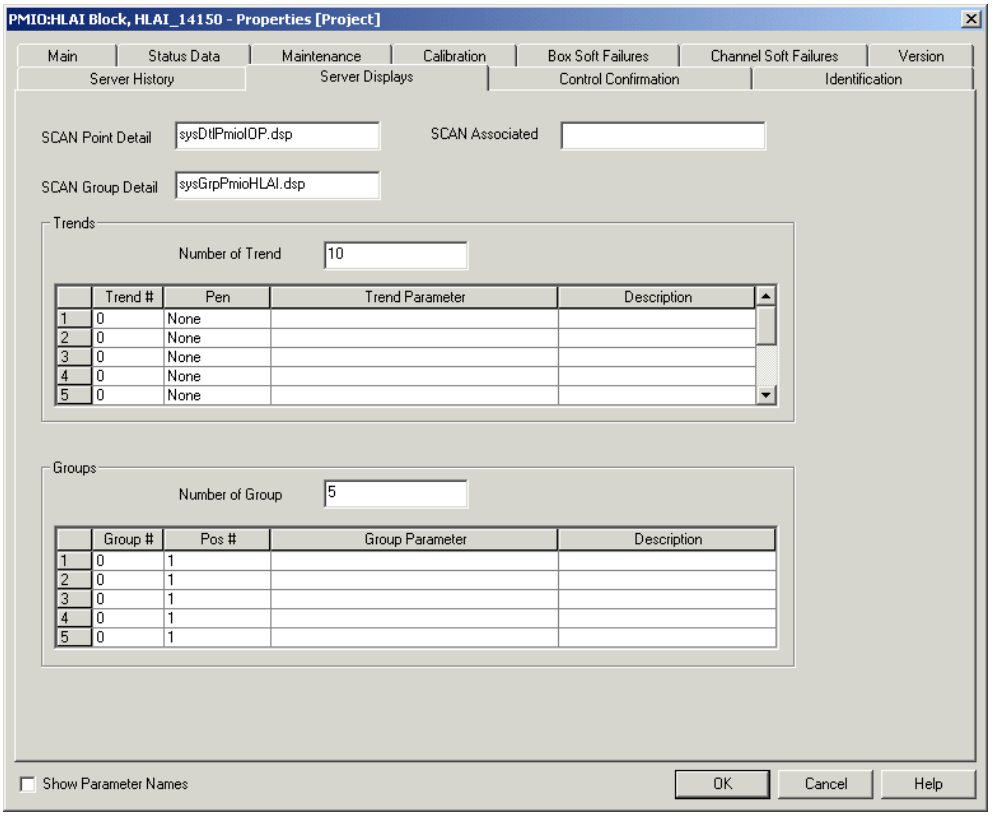


TIP

If you have a Distributed Server Architecture (DSA), you must enter the Control Area assignment for this Server. (Note that area code assignments are made through the Station application.) If you do not have a DSA, you can skip this field if area is not enabled through the Station application.

11	Use the on-line help as a guide to complete the configuration entries on this tab. Click the Server Displays tab.	Calls up the Server Displays configuration form.

Step	Action	Result



Step	Action	Result
12	Use the on-line help as a guide to complete the configuration entries on this tab. Click the OK button.	Closes the form and creates the block icon along with its Channel blocks in Project tab. The following IOP block icons and Channel blocks can be created: • HLAI100 block icon along with its Analog Input Channel block • HLAIHART block icon along with its Analog Input Channel block • LLMUX block ison along with its Analog Input Channel block • AO16 block icon along with its Analog Output Channel block • AO16HART block icon along with its Analog Output Channel block • DI24V block icon along with its Digital Input Channel • DO32 block icon along with its Digital Output Channel block • STIMV block icon along with its Analog Input Channel blocks
13	This completes the creation of one of the following: • HLAI100 IOP • HLAIHART IOP • LLMUX IOP • AO16 IOP • AO16HART IOP • DI IOP • DO IOP • STIMV IOP	Other data on the configuration form is unavailable in Project mode. See the on-line help for information about the other parameters.
	Note: To configure IO Channel blocks assigned to the HLAIHART IOP or AO16HART	

Step	Action	Result
	IOP, go to PM I/O HART Device Configuration in the *HART Implementation Guide*.	

Creating PM I/O HART High Level Analog Input IOP

Use the following procedure as a general guide for creating and configuring a function block for a HART High Level Analog Input (HLAIHART) type Input/Output Processor (IOP).

Prerequisites:

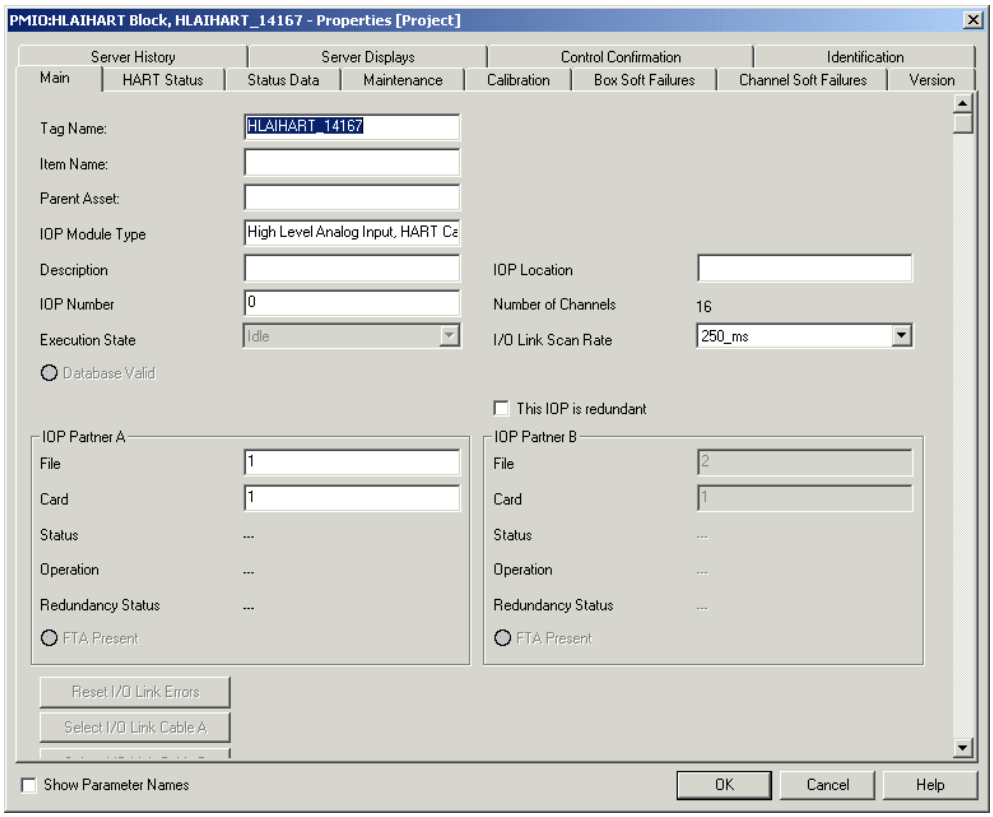
- Control Builder is running
- Tree windows are open

Considerations:

- The following procedure uses the menu method of creation but the Drag and Drop Method described above for Input/Output Modules can also be used
- All illustrations used in the procedure are for example purposes only.

Step	Action	Result
1	Click File->New->I/O Modules ->PMIO->HLAIHART - High Level Analog Input, HART Capable, 16 ch.	Calls up the HLAIHART Block configuration form with Tag Name field highlighted.

Step	Action	Result



To complete the configuration of the module, refer to: *Completing the configuration form*.

Creating PM I/O Low Level Analog Multiplexer Input IOP

Use the following procedure as a general guide for creating and configuring a function block for a Low Level Analog Multiplexer Input (LLMUX) type Input/Output Processor (IOP).

Prerequisites:

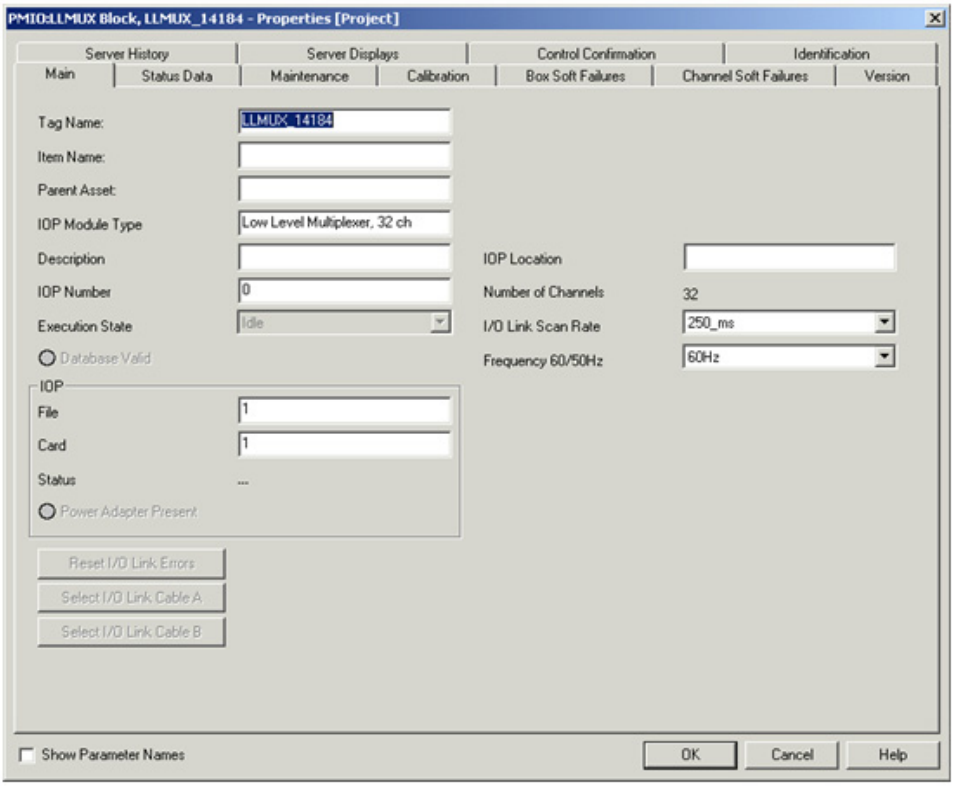
- Control Builder is running
- Tree windows are open

Considerations:

- The following procedure uses the menu method of creation but the Drag and Drop Method described above for Input/Output Modules can also be used
- All illustrations used in the procedure are for example purposes only.

Step	Action	Result
1	Click File->New->I/O Modules->PMIO-> LLMUX - Low Level Multiplexer, 32 ch.	Calls up the LLMUX Block configuration form with Tag Name field highlighted.

Step	Action	Result



To complete the configuration of the module, refer to: *Completing the configuration form*.

Creating PM I/O Analog Output IOP

Use the following procedure as a general guide for creating and configuring a function block for an Analog Output (AO) type Input/Output Processor (IOP).

Prerequisites:

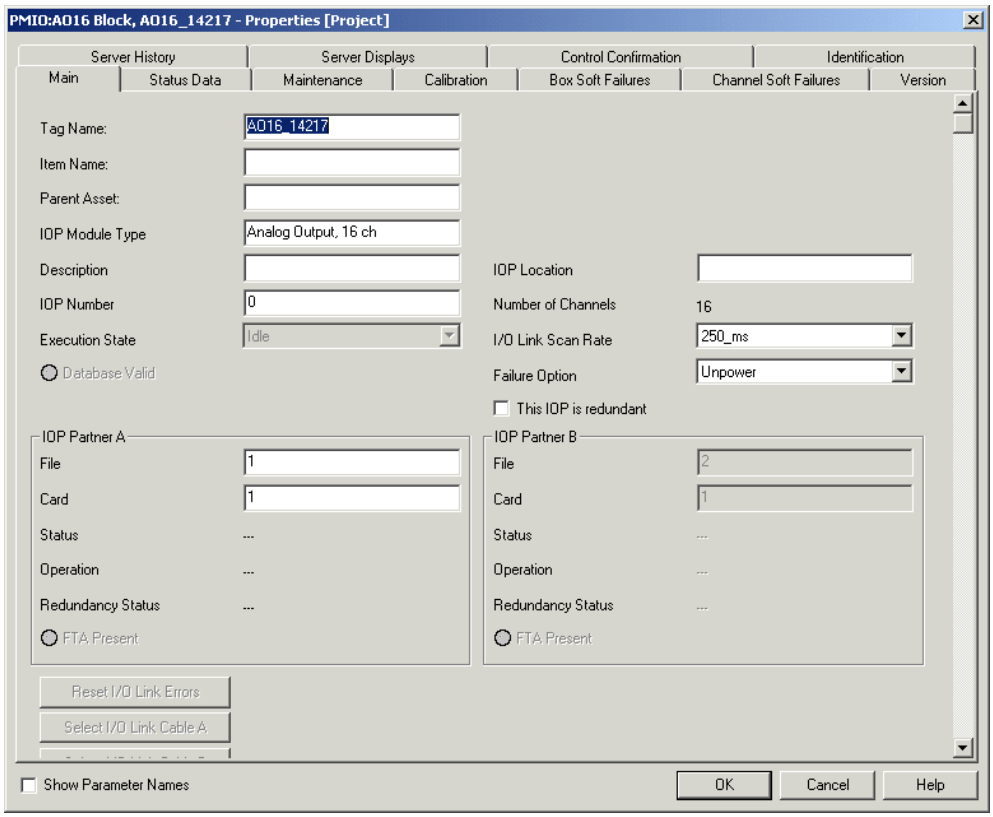
- Control Builder is running
- Tree windows are open

Considerations:

- The following procedure uses the menu method of creation but the Drag and Drop Method described above for Input/Output Modules can also be used
- All illustrations used in the procedure are for example purposes only.

Step	Action	Result
1	Click File->New->I/O Modules->PMIO-> AO16 - Analog Output, 16 ch.	Calls up the AO16 Block configuration form with Tag Name field highlighted.

Step	Action	Result



To complete the configuration of the module, refer to: *Completing the configuration form*.

Creating PM I/O HART Analog Output IOP

Use the following procedure as a general guide for creating and configuring a function block for a HART Analog Output (AO16HART) type Input/Output Processor (IOP).

Prerequisites:

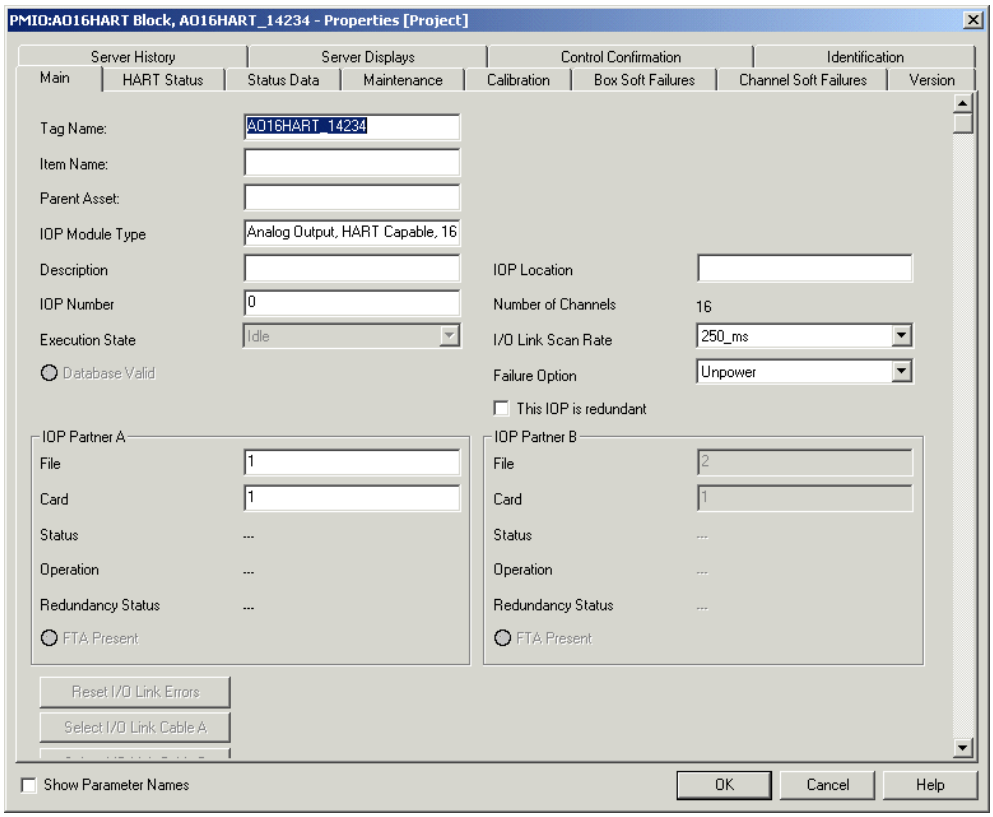
- Control Builder is running
- Tree windows are open

Considerations:

- The following procedure uses the menu method of creation but the Drag and Drop Method described above for Input/Output Modules can also be used
- All illustrations used in the procedure are for example purposes only.

Step	Action	Result
1	Click File->New->I/O Modules-> PMIO-> AO16HART - Analog Output, HART Capable, 16 ch.	Calls up the AO16HART Block configuration form with Tag Name field highlighted.

Step	Action	Result



To complete the configuration of the module, refer to: *Completing the configuration form*.

Creating PM I/O Digital Input IOP

Use the following procedure as a general guide for creating and configuring a function block for a Digital Input (DI) type Input/Output Processor (IOP).

Prerequisites:

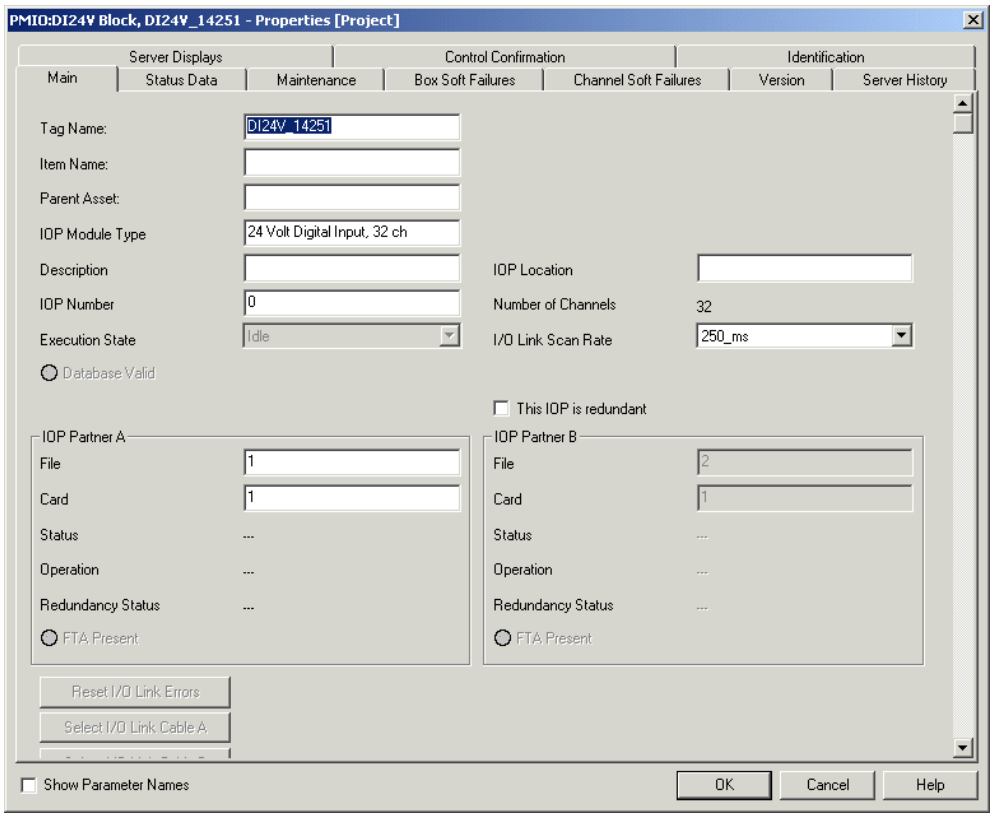
- Control Builder is running
- Tree windows are open

Considerations:

- The following procedure uses the menu method of creation but the Drag and Drop Method described above for Input/Output Modules can also be used
- All illustrations used in the procedure are for example purposes only.

Step	Action	Result
1	Click File->New->I/O Modules->PMIO-> DI24V - 24 Volt Digital Input, 32 ch.	Calls up the DI24V Block configuration form with Tag Name field highlighted.

Step	Action	Result



To complete the configuration of the module, refer to: *Completing the configuration form*.

Creating PM I/O Digital Output IOP

Use the following procedure as a general guide for creating and configuring a function block for a Digital Output (DO) type Input/Output Processor (IOP).

Prerequisites:

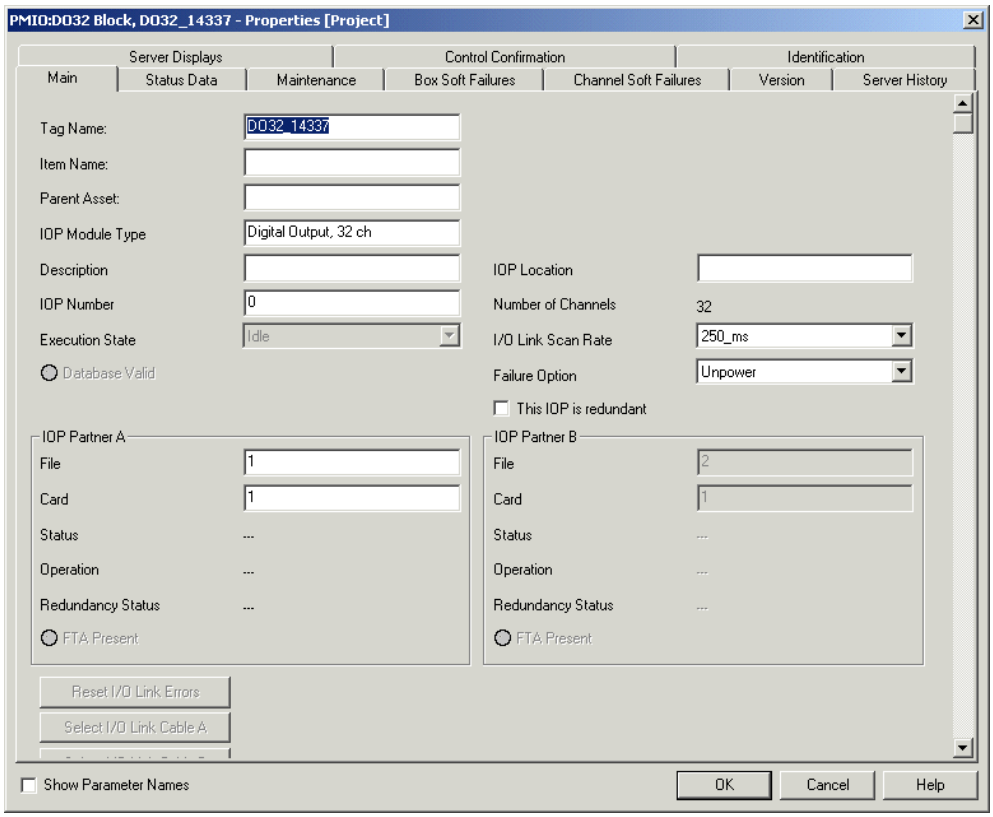
- Control Builder is running
- Tree windows are open

Considerations:

- The following procedure uses the menu method of creation but the Drag and Drop Method described above for Input/Output Modules can also be used
- All illustrations used in the procedure are for example purposes only.

Step	Action	Result
1	Click File->New->I/O Modules->PMIO-> DO32 - Digital Output, 32 ch.	Calls up the DO32 Block configuration form with Tag Name field highlighted.

Step	Action	Result



To complete the configuration of the module, refer to: *Completing the configuration form*.

Creating PM I/O STIMV IOP

Use the following procedure as a general guide for creating and configuring a function block for a Smart Transmitter Interface Multi-Variable (STIMV) type Input/Output Processor (IOP).

Prerequisites:

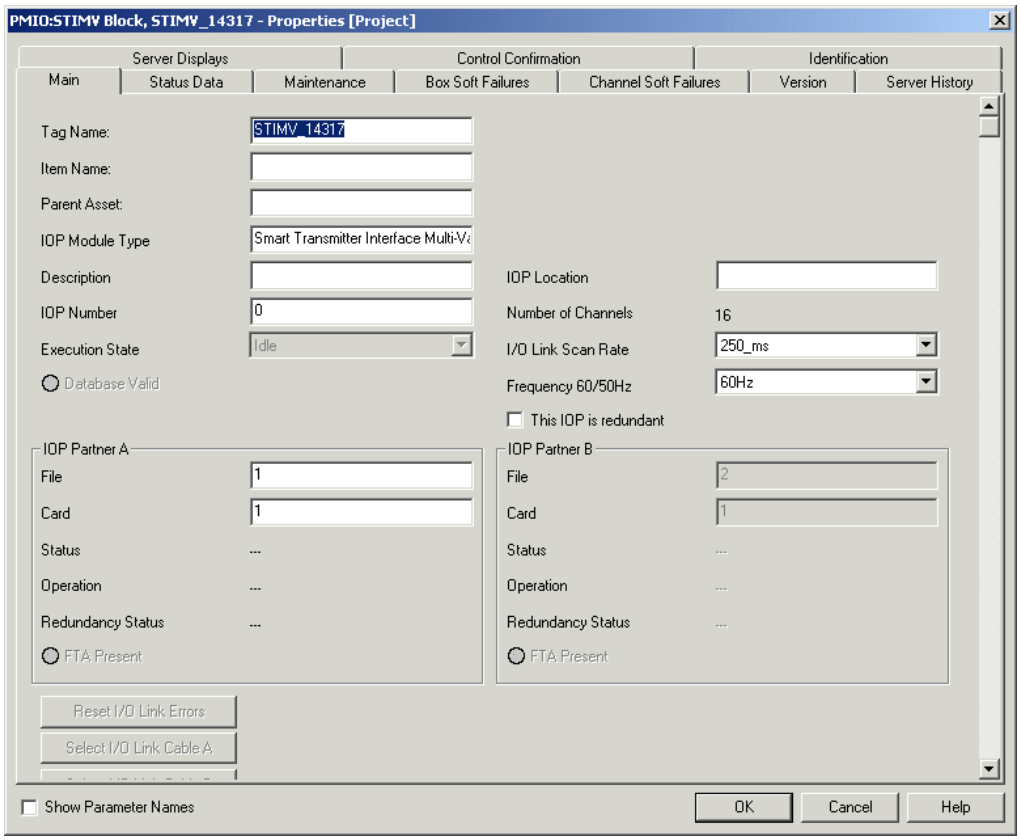
- Control Builder is running
- Tree windows are open

Considerations:

- The following procedure uses the menu method of creation but the Drag and Drop Method described above for Input/Output Modules can also be used
- All illustrations used in the procedure are for example purposes only.

Step	Action	Result
1	Click File->New->I/O Modules->PMIO-> STIMV - Smart Transmitter Interface Multi-Variable, 16 ch.	Calls up the STIMV Block configuration form with Tag Name field highlighted.

Step	Action	Result



To complete the configuration of the module, refer to: *Completing the configuration form*.

Creating a control module

This section includes information that you can reference for:

Topic	Link
Creating and saving a control module	Click here
Copying control modules	Click here
Assigning CMs and IOMs to CEE	Click here
Assigning IOPs to IOLINK	Click here
Creating an instance of a basic function lock	Click here
Assigning PMIO IOC block to IOP	Click here
Unassigning PMIO IOC block from IOP	Click here
Copy a function block	Click here
Move function blocks within a chart	Click here
Delete a function block	Click here
Use the Parameters Configuration form	Click here
Setting system preferences	Click here
Setting user preferences	Click here
Connecting and disconnecting blocks	Click here
Associating I/O channels to I/O Modules (IOMs)	Click here
Inserting OLE objects into charts	Click here
User Server Scripting in Control Builder	Click here
Printing from Control Builder	Click here
Using the Regulatory Loop Wizard	Click here
Exporting function block configurations	Click here
Importing function block configurations	Click here
Use Module Hierarchy	Click here

Creating and saving a control module

To create a Control Strategy, a Control Module must be created and function blocks inserted and connected. The following graphic shows Control Builder with a Control Module chart shown in the Control Drawing area.

Prerequisites:

- Control Builder is running
- Tree windows are open

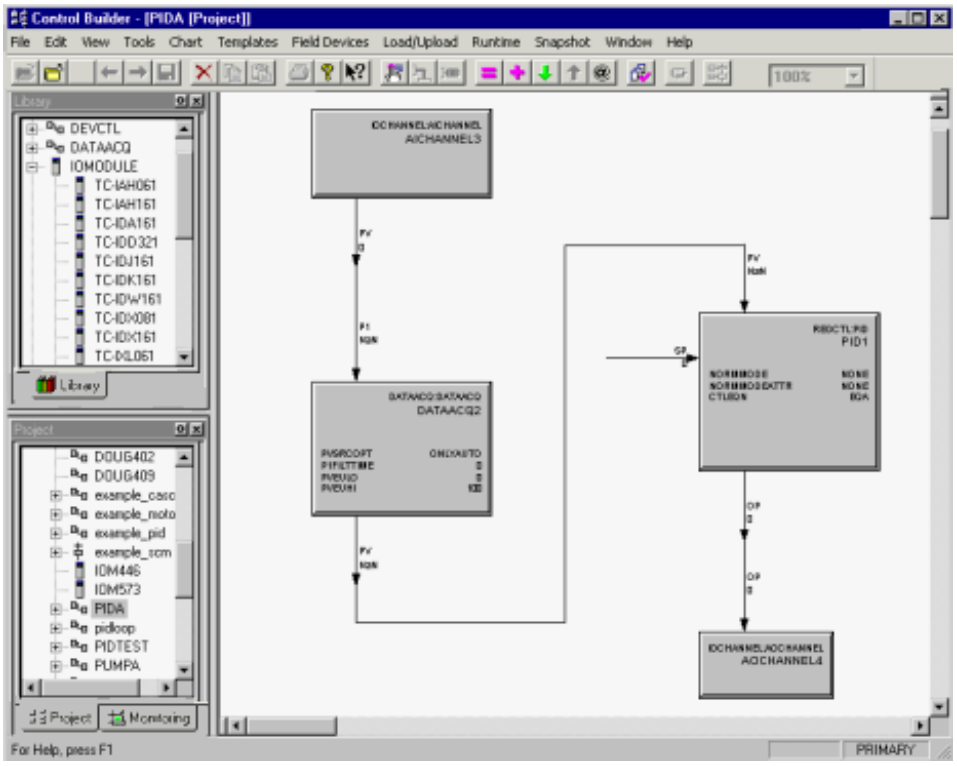
Considerations:

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



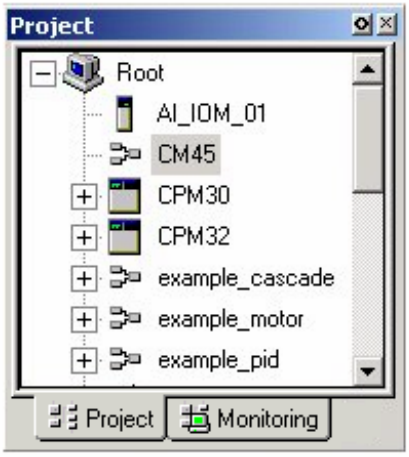
WARNING

All edits done on project-related objects must be reloaded to the controller before those edits can be seen in the controller. See *Control Strategy Loading* Overview for information on how to load control strategy objects.



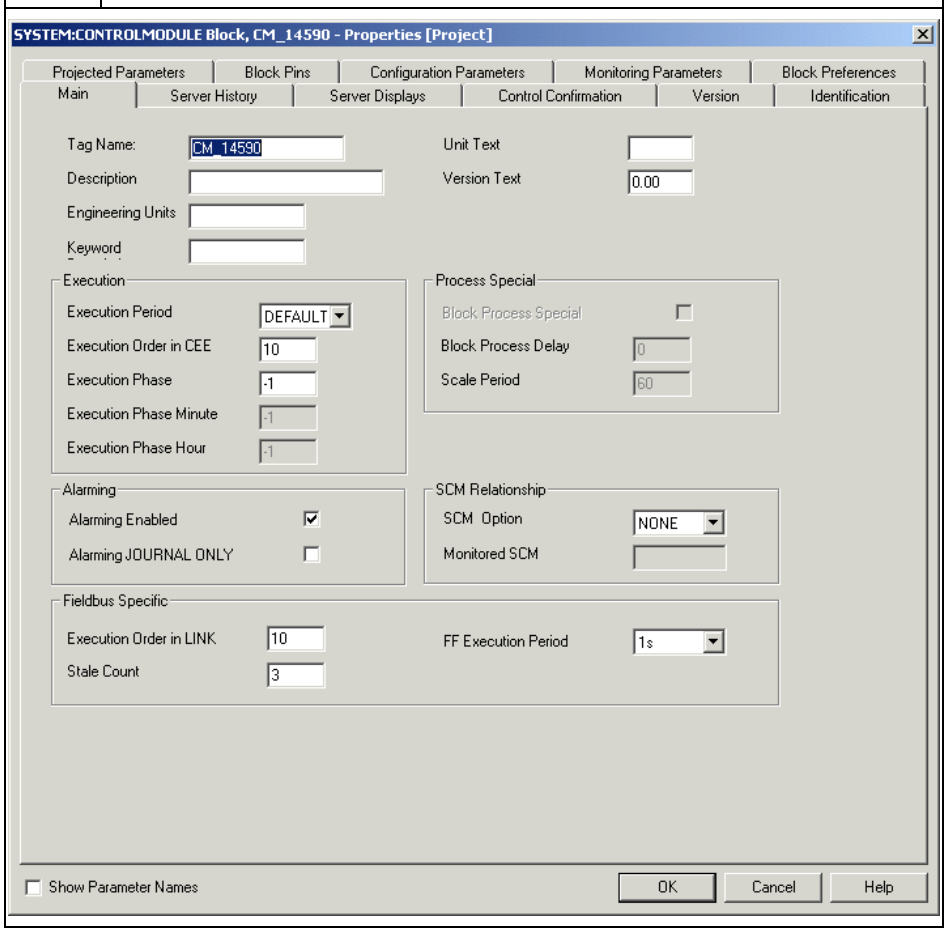
Example of Control Builder CM frame

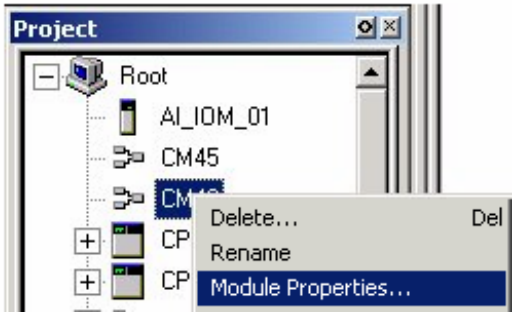
Use the following procedure to create and save a Control Module.

Step	Action
1	Click File -> New -> Control Module A blank Control Module is opened up in the Control Drawing area. • The new Control Module appears under the Root Project Tree. Control Module names are sequentially numbered (for example, CM30, CM31, etc.). • The new Control Module is automatically saved to your harddrive.  ATTENTION: Instead of using step 1 to create a Control Module, you can alternately use either step 1A, 1B or 1C which follow.
1A	(ALTERNATE 1A) Set up Control Builder with both the Library and Project views visible. See Opening and navigating a tree window**.** • Click on and expand the SYSTEM Library in the Library Tree. • Drag and drop a CONTROLMODULE block from the System Library onto the Project Root. The new Control Module appears under the Root Project Tree. Control Module names are sequentially numbered (for example, CM30, CM31, etc.). The new Control Module is automatically saved to your harddrive.

Step	Action
1B	(ALTERNATE 1B) Set up Control Builder with both the Library and Project views visible. See Opening and navigating a tree window. • Double-click on CONTROLMODULE in the Library tab A blank Control Module is opened up in the Control Drawing area. The new Control Module appears under the Root Project Tree. Control Module names are sequentially numbered (for example, CM30, CM31, etc.). The new Control Module is automatically saved to your harddrive.
1C	(ALTERNATE 1C) Copy a Control Module. See Copying Control Modules.
2	Select Tools -> Module Properties… or double-click with the mouse cursor located anywhere inside the chart to open the Control Module Parameter Configuration form for input.

Step	Action


SYSTEM:CONTROLMODULE Block, CM_14590 - Properties [Project]
Projected Parameters
Block Pins
Configuration Parameters
Monitoring Parameters
Block Preferences
Main
Server History
Server Displays
Control Confirmation
Version
Identification
Tag Name:
CM_14590
Description
Engineering Units
Keyword
Unit Text
Version Text
0.00
Execution
Execution Period
DEFAULT
Execution Order in CEE
10
Execution Phase
-1
Execution Phase Minute
-1
Execution Phase Hour
-1
Process Special
Block Process Special
Block Process Delay
0
Scale Period
60
Alarming
Alarming Enabled
Alarming JOURNAL ONLY
SCM Relationship
SCM Option
NONE
Monitored SCM
Fieldbus Specific
Execution Order in LINK
10
Stale Count
3
FF Execution Period
1s
Show Parameter Names
OK
Cancel
Help

Step	Action
	Note: If the Control Module's chart is NOT open in the Control Drawing area, right-click on the new Control Module in the Project Tree to display the selection options and click on Configure Module Parameters to open the Control Module Parameter Configuration form for input. 
3	Enter a new Control Module name in the Name field along with a description in the Description field.
4	Using the F1 key to access context-sensitive Help, fill in the remaining fields as required.
5	Click OK. Configuration Form closes.
6	If necessary, double-click on the newly-named Control Module in the Project Tree to open it. The new name appears at the top of the Control Module drawing when the Control Drawing opens.
7	Click File -> Save to save any additional changes you make to the Control Module before closing.
8	Click File -> Close to close the chart.

Copying control modules

Prerequisites:

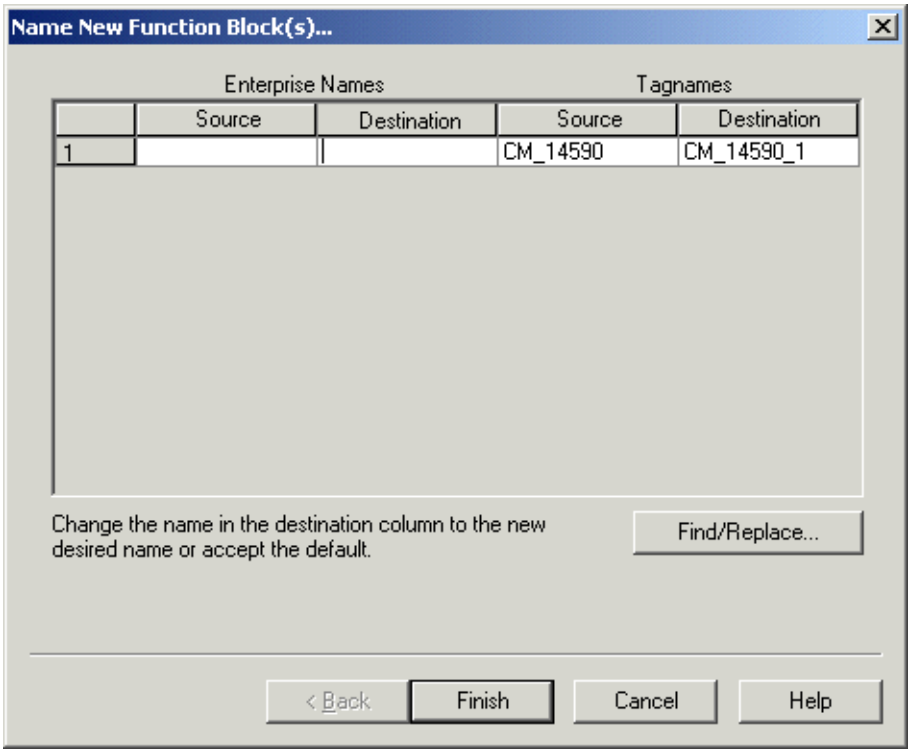
- Control Builder is running
- Tree windows are open

Considerations:

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

Perform the steps in the table below to copy an existing control module.

Step	Action	Result
1	From the Project Tree, select the desired Control Module block to be copied.	Selected Control Module block is highlighted on the Project Tree.
2	Click Edit -> Copy. Alternate method: • Click <Ctrl>-C.	Selected Control Module block is saved to Control Builder clipboard and Name New Function Block(s)… dialog appears.
3	Change the Control Module block's Tagname in the Destination column of the Name New Function Block(s) dialog to a desired name or accept the default name.	The copied Control Module block is assigned a new name. Note: You may opt to keep the default name which is simply the original name with a number appended to it.
4	Click Next to proceed to the next dialog page (if appropriate) and enter new names as prompted to resolve any existing connections and/or associations.	If the CM contains connections to outside blocks, an additional dialog page appears which is used to resolve any existing connections and/or associations.
5	Click Finish	Copied Control Module block with newly-designated name is pasted onto the Project Tree.



Name New Function Block(s) dialog

Assigning CMs and IOMs to CEE

Use the following procedure as a general guide to assign configured Control Modules (CMs) and I/O Modules (IOMs) to configured Control Execution Environments (CEEs) for Control Processor Module (CPM), Application Control Environment (ACE), or Simulation Control Environment (SCE) blocks.

Prerequisites:

- Control Builder is running
- Tree windows are open

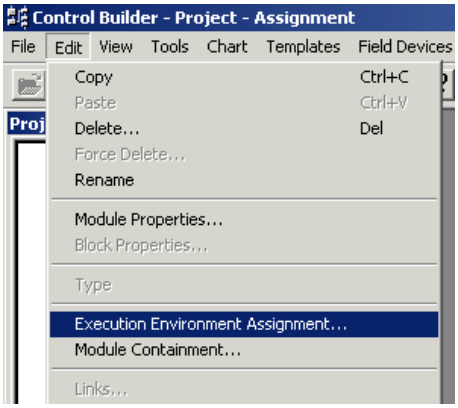
Considerations:

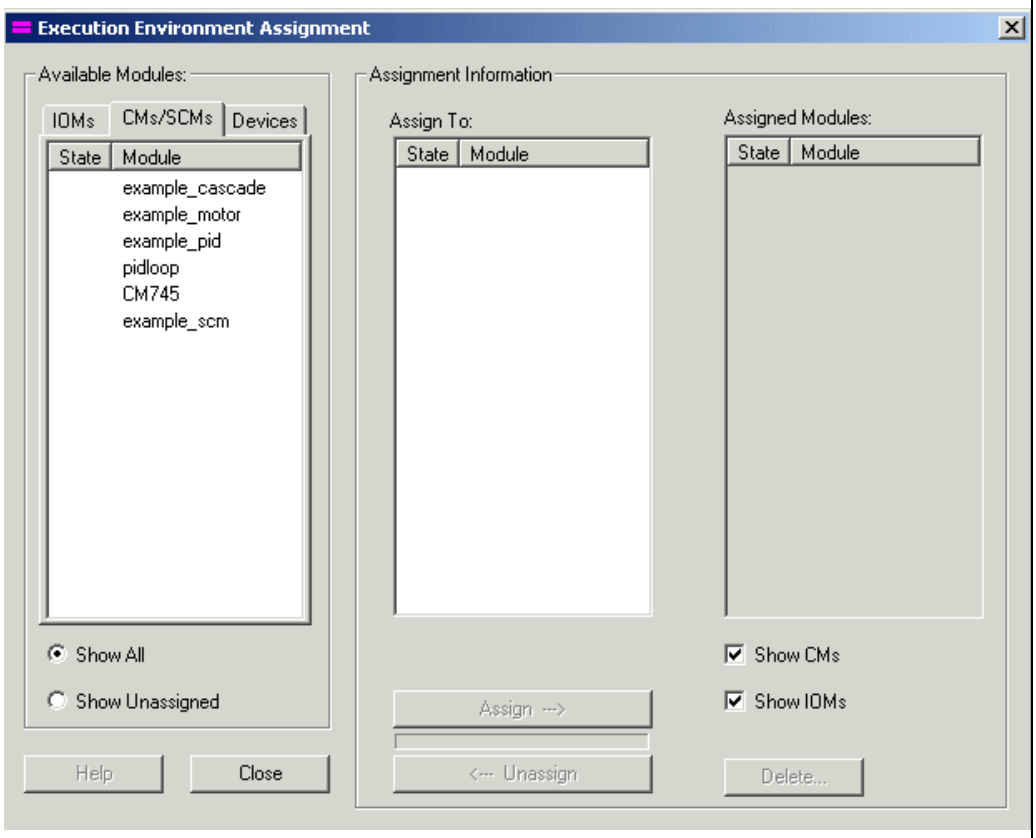
- All illustrations used in the procedure are for example purposes only.

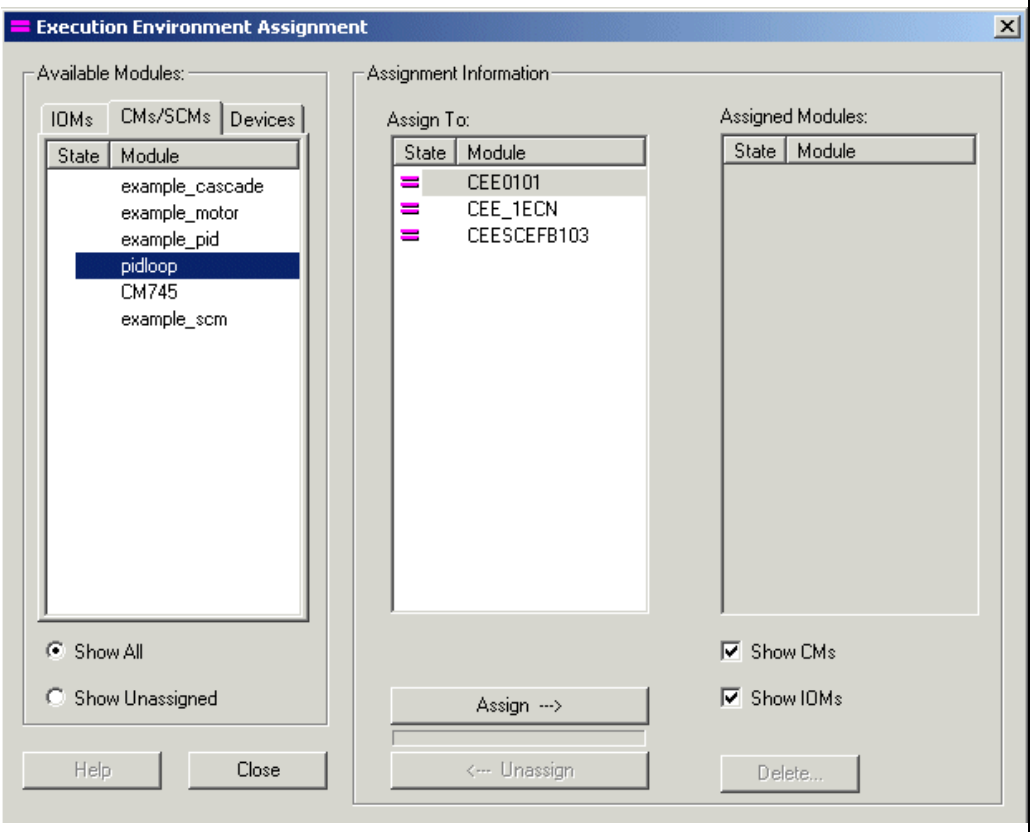
ATTENTION

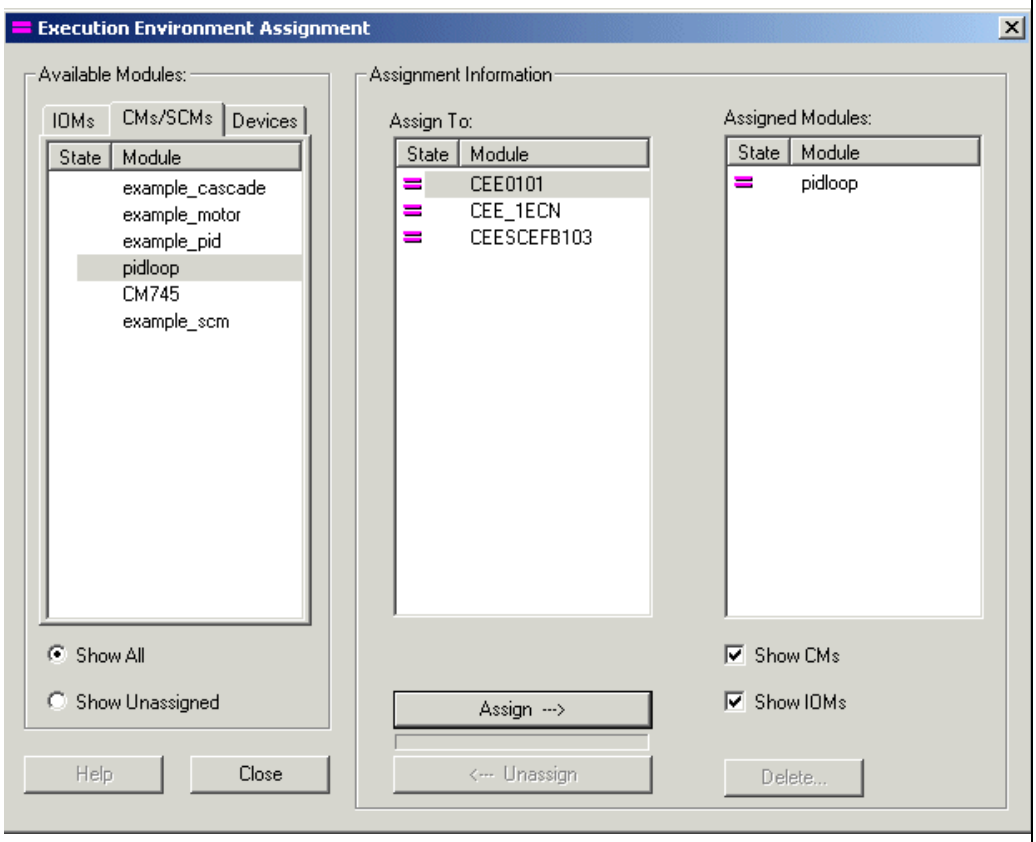
- Before Control Builder allows you to associate an IOM to an IOCHANNEL block, it checks to make sure that the CM and IOM are assigned to the same CEE.
- All edit windows (such as CM charts) must be closed before proceeding with this procedure or a lock contention may occur. To resolve these types of lock contentions, close the open edit window (such as the CM chart) and attempt to perform the procedure again.

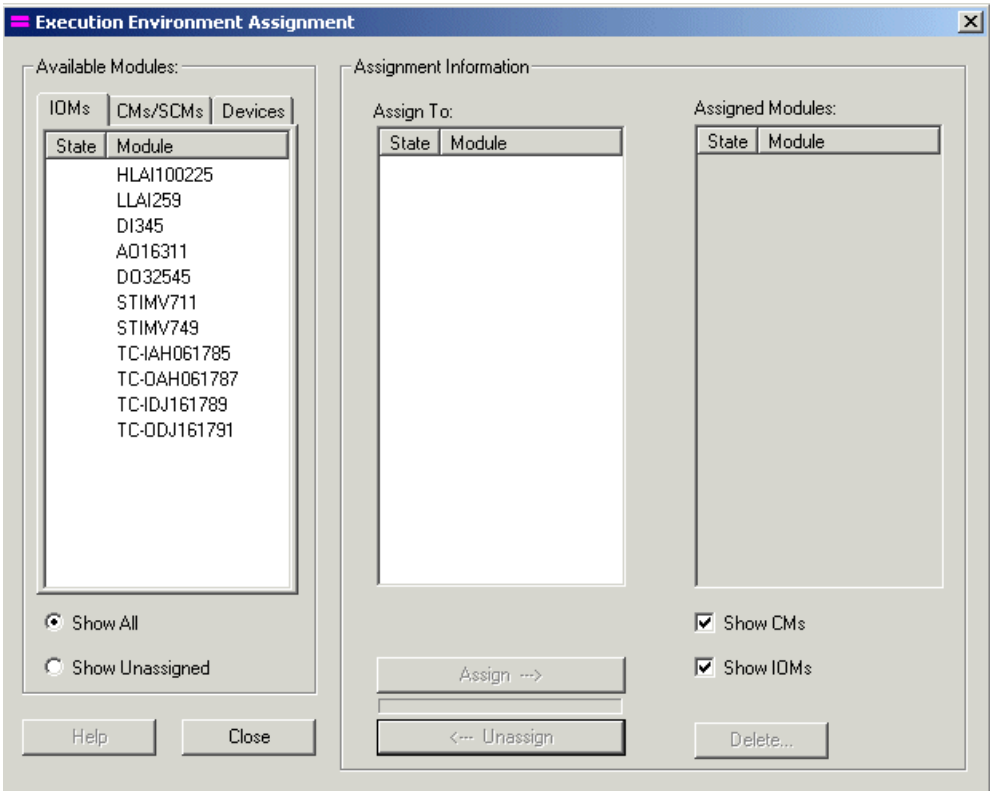
Step	Action	Result
1	Click Edit->Execution Environment Assignment. Or, click  assignment button in the toolbar.	Calls up Execution Environment Assignment dialog box. (There is no set default state for the dialog, so it may come up with different active fields than shown below.)

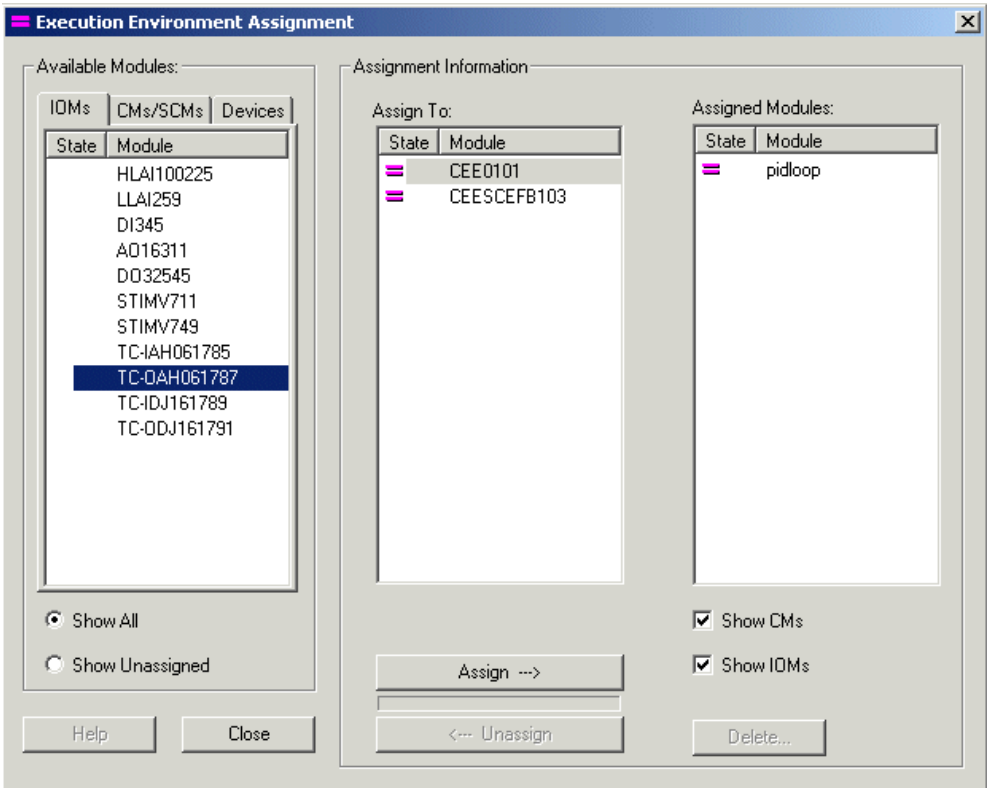


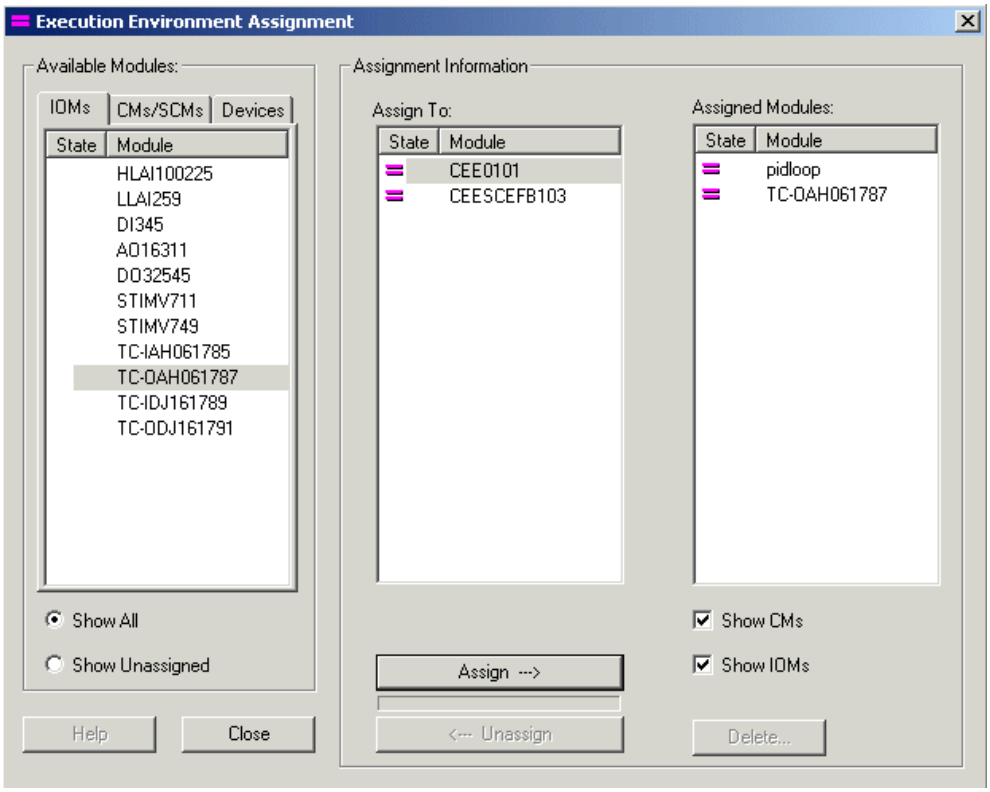
Step	Action	Result
		
	TIP Can use common <Shift> plus click and <Control> plus click actions to select multiple items in Available Modules and Assigned Modules lists.	
2	With CMs/SCMs tab selected, click listed CM to be assigned to a CEE.	Highlights selection and configured CEEs appear in the Assign To list.

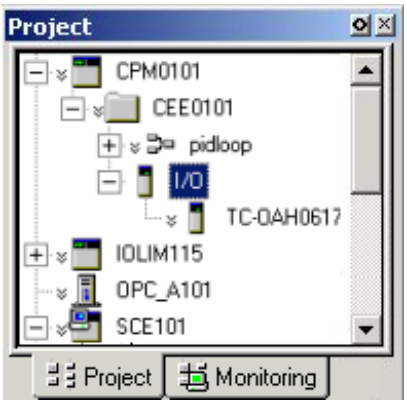
Step	Action	Result
		
3	Accept default CEE selection or click desired CEE in list.	Be sure correct CEE is selected in list.
4	Click the  assign button.	Selected CM is assigned to selected CEE and appears in the Assigned Modules list.

Step	Action	Result
		
5	Click IOMs tab in Available Modules section.	Configured IOMs/IOPs appear in list.

Step	Action	Result
		
6	Click listed IOM to be assigned.	Highlights selection and applicable CEEs appear in Assign To list.

Step	Action	Result
		
7	Accept default CEE selection or click desired CEE in list.	Be sure correct CEE is selected in list.
8	Click the Assign ---> assign button.	Selected IOM is assigned to selected CEE and appears in the Assigned Modules list.

Step	Action	Result
		
9	Repeat Steps 2 to 4 to assign other CMs/SCMs. Or, repeat Steps 5 to 8 to assign other IOMs.	Complete CM, SCM, and IOM assignments.

Step	Action	Result
10	Click  Close button.	Closes dialog box and assigned components now appear in CEE folder in Project tab. 

Assigning IOPs to IOLINK

Use the following procedure as a general guide to assign configured Input/Output Processors (IOPs) to configured Input/Output Link (IOLINK) for given Input/Output Link Interface Module (IOLIM) blocks. The illustrations used in the following procedure are for example purposes only.

Prerequisites:

- Control Builder is running
- Tree windows are open

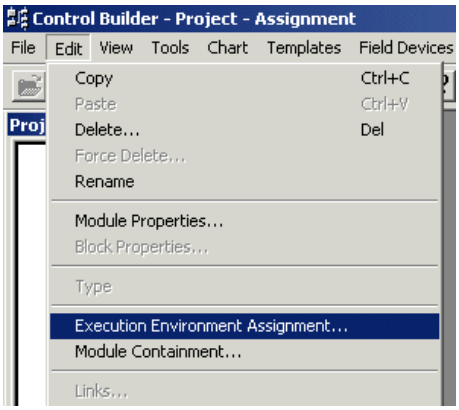
Considerations:

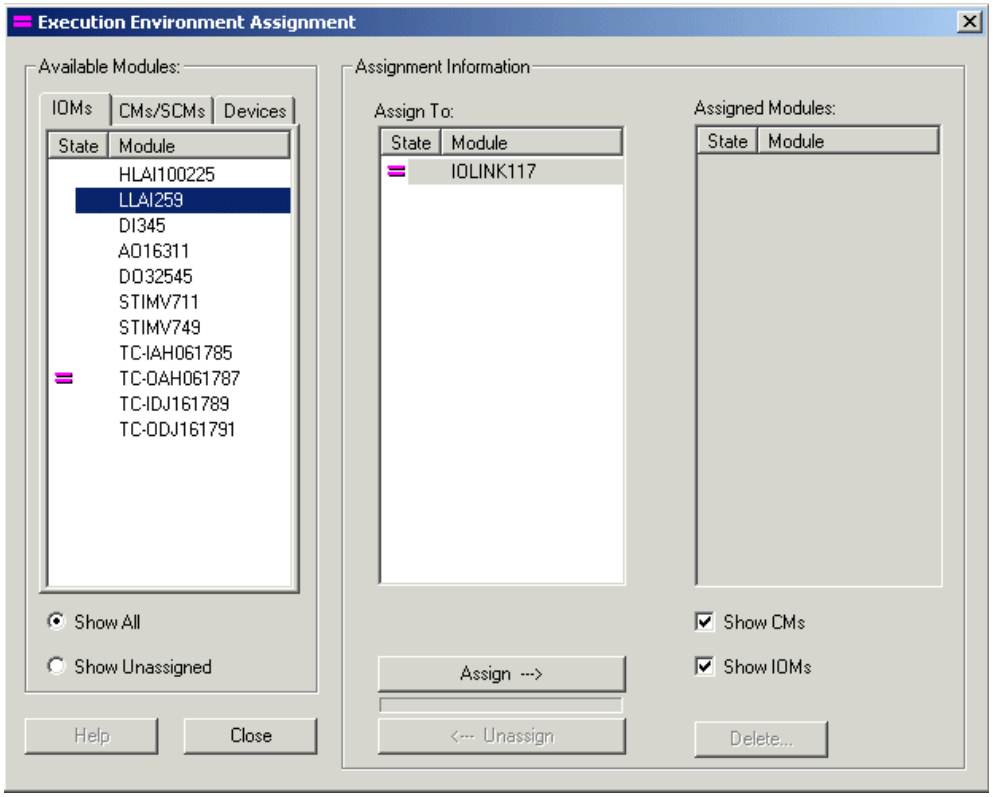
- All illustrations used in the procedure are for example purposes only.

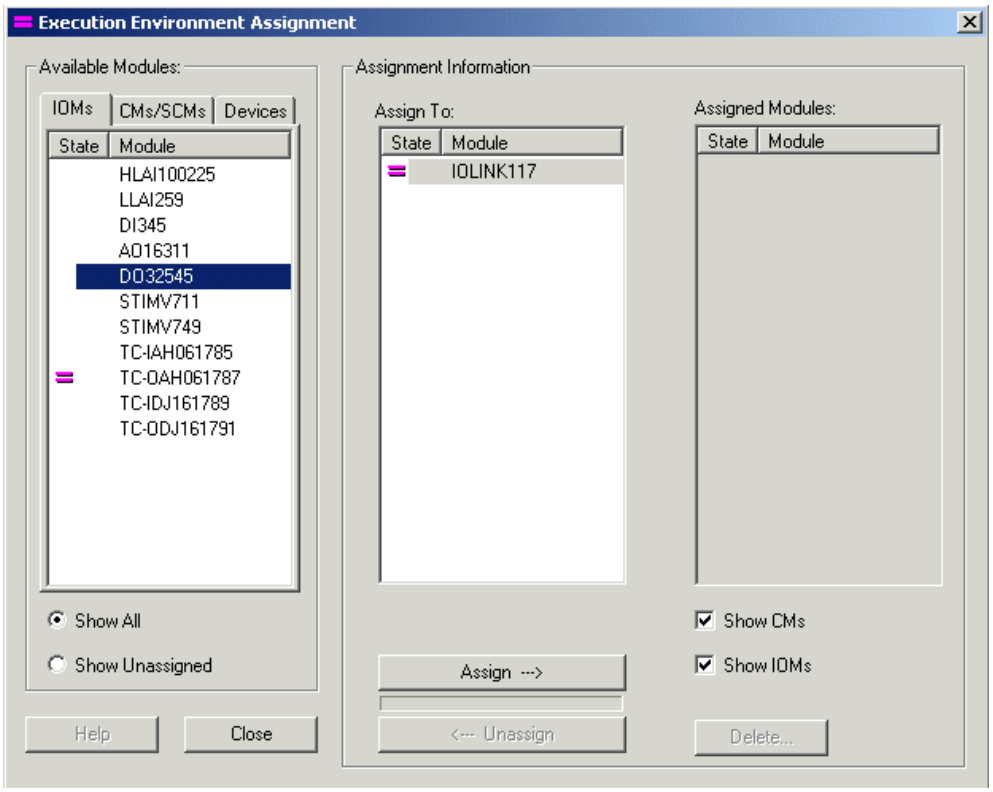
ATTENTION

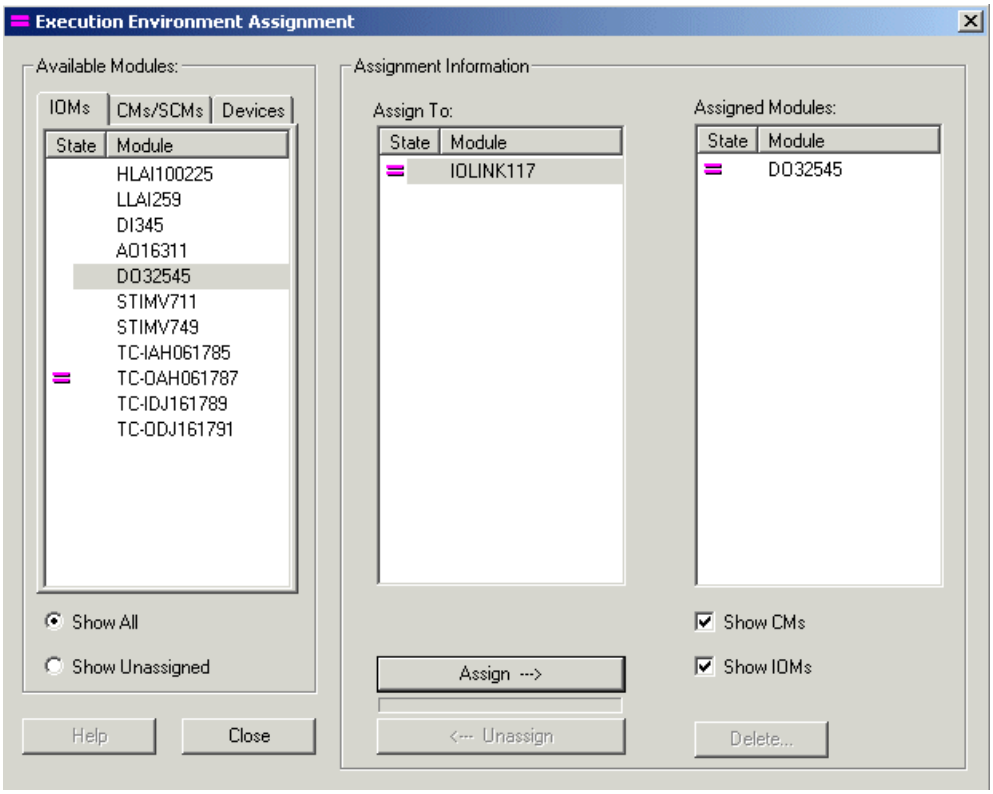
- You must assign Input/Output Processors (IOP) to an I/O Link before you can assign Input/Output Channels (IOC) to a given IOP.
- Configure IOP first to avoid duplicate IOP number error message when attempting to assign an IOP to an IOLINK. System checks for unique IOP identification before assigning it to the IOLINK.

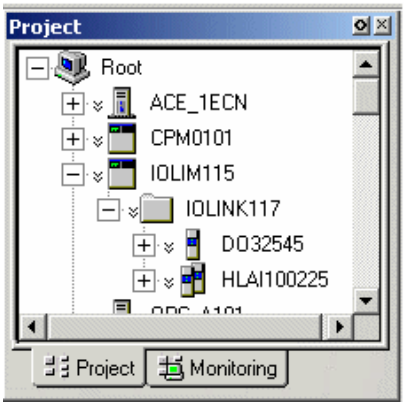
Step	Action	Result
1	Click Edit->Execution Environment Assignment. Or, click  assignment button in the toolbar.	Calls up Execution Environment Assignment dialog box. (There is no set default state for the dialog, so it may come up with different active fields than shown below.)

Step	Action	Result
		

Step	Action	Result
		
	TIP Can use common <Shift> plus click and <Control> plus click actions to select multiple items in Available Modules and Assigned Modules lists.	
2	With IOMs tab selected, click desired IOP to be assigned to given IOLINK.	Highlights selection and configured IOLINKs appear in the Assign To list.

Step	Action	Result
		
3	Accept default IOLINK selection or click desired IOLINK in list.	Be sure correct IOLINK is selected in list.
4	Click the Assign ---> assign button.	Selected IOP is assigned to selected IOLINK and appears in the Assigned Modules list.

Step	Action	Result
		
5	Repeat Steps 2 to 4 to assign other IOPs.	Complete IOP assignments.

Step	Action	Result
6	Click  Close button.	Closes dialog box and assigned components now appear in IOLINK folder in Project tab. 

Creating an instance of a basic function block

Perform the steps below to create an instance of a basic function block. Blocks appear as Block Symbols on the Control Drawing.

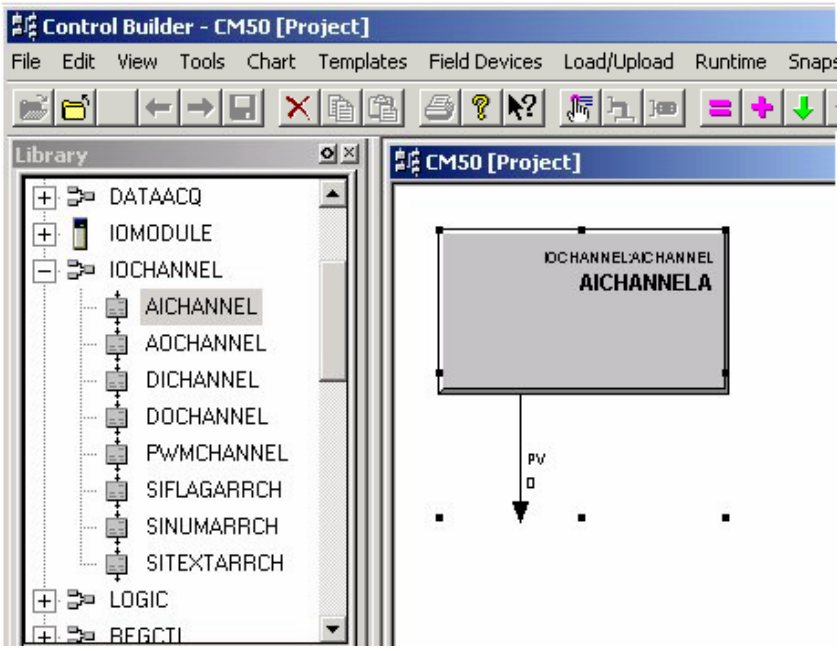
Prerequisites:

- Control Builder is running
- Tree windows are open

Considerations:

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

Step	Action	Result
1	Make sure a Control Module (CM) chart is open. • Double-click the CM in the Project Tree to open your CM chart so function blocks may be added. • See Creating and Saving a Control Module.	The Control Module (CM) chart is open so that function blocks may be added.
2	From the appropriate Library Tree group, drag and drop the desired block (or blocks) to the Control Module Control Drawing (such as an AICHANNEL block from the IOCHANNEL group).	Desired block is shown in reverse video on the tree. The new function block appears on the Control Module chart.
	**TIP** • You can drag and drop PMIO Input/Output Channel (IOC) blocks directly from IOPs added to the Project tab to the Control Module. • You can not add PMIO IOC blocks to Control Modules already assigned to a Fieldbus Interface Module (FIM).	
3	Repeat Step 2 as many times as necessary to create the desired blocks for your control strategy.	



Control Module chart with an AICHANNEL block

Assigning PMIO IOC block to IOP

Use the following procedure as a general guide to assign a Process Manager I/O, Input/Output Channel (IOC) block to an I/O Processor (IOP).

Prerequisites:

- This procedure assumes that an instance of an IOC block was created from the PMIO library in a Control Module (CM) as noted in the previous section
- Control Builder is running
- Tree windows are open

Considerations:

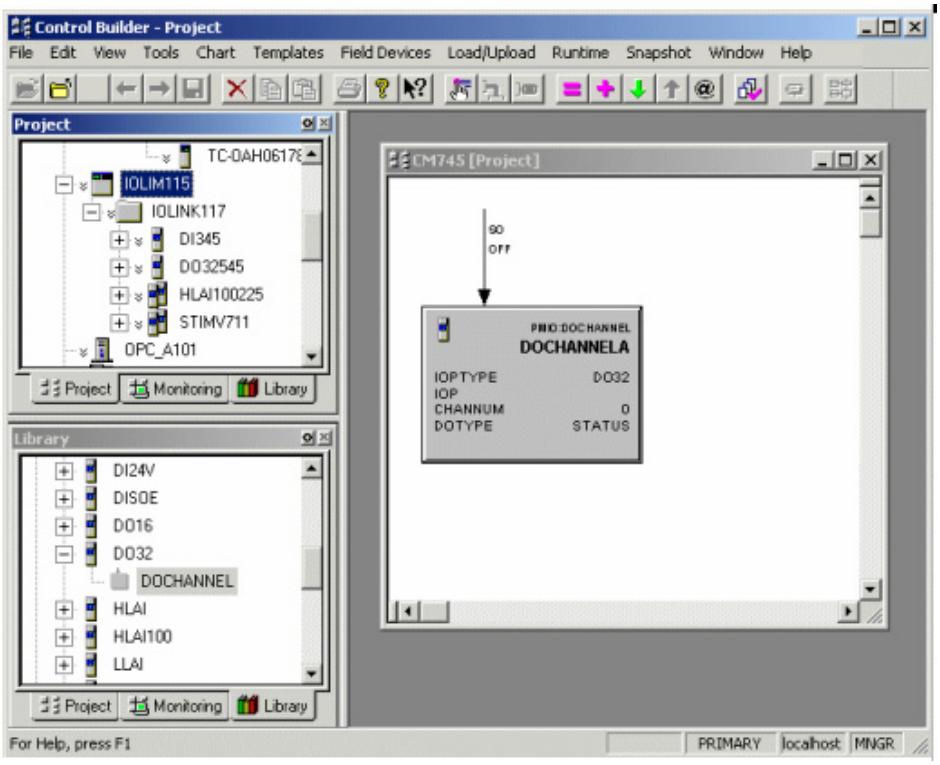
- All illustrations used in the procedure are for example purposes only.

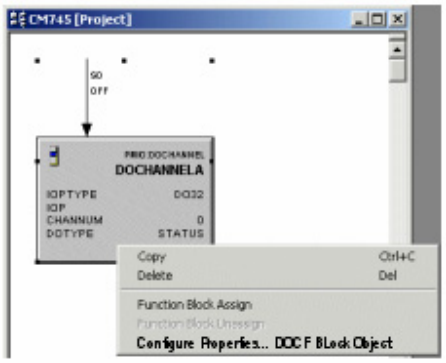


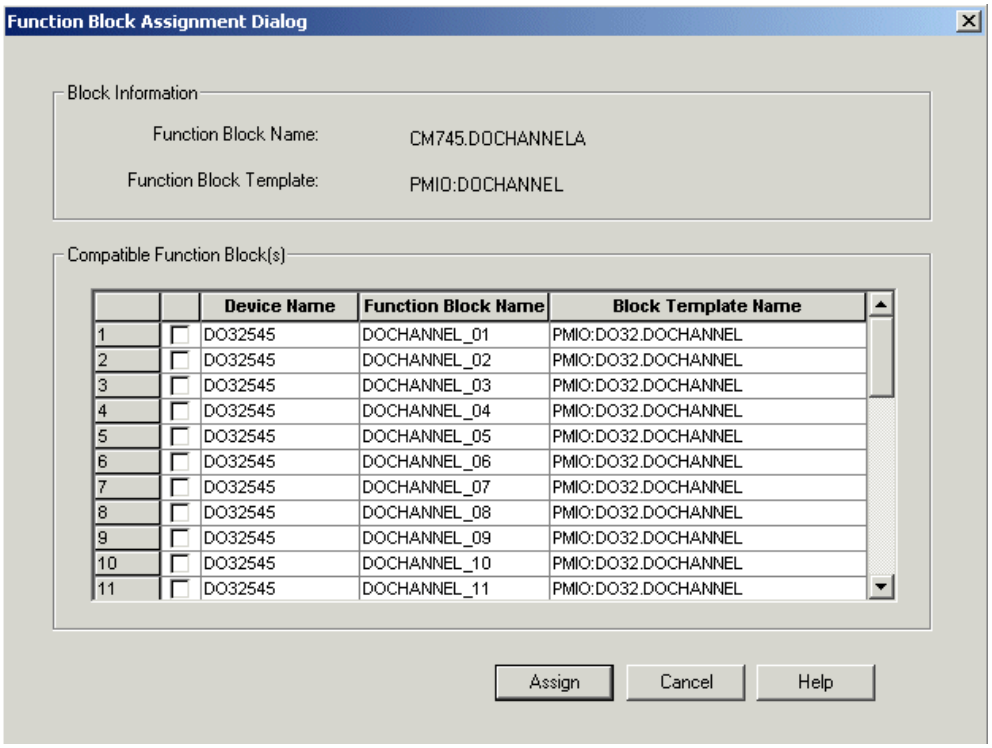
TIP

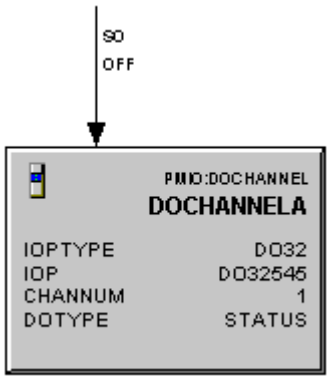
This procedure is only required when an IOC is added from an IOP in the PMIO directory in the Library tab or when an "unassigned" IOC needs to be re-assigned.

Step	Action	Result
1	Double click given CM icon in Project tree.	Opens CM in control chart area.



Step	Action	Result
2	Right-click PMIO IOC block.	Opens shortcut menu. 
3	Select Function Block Assign.	Opens Function Block Assignment Dialog that list compatible IOCs associated with IOPs assigned to an IOLINK.

Step	Action	Result
		
4	Click check box for desired channel in given IOP.	Selects channel for assignment.
	ATTENTION You can **not** assign an IOC to an IOP channel that is currently loaded in the IOLINK and appears in the Monitoring tree.	

Step	Action	Result
5	Click the Assign Assign button.	Close dialog box and assigns IOC to selected channel in named IOP. 
6	Close the CM and click Yes.	Closes CM and saves changes.
7	Repeat procedure as required to assign other PMIO library IOCs.	

Unassigning PMIO IOC block from IOP

Use the following procedure as a general guide to unassign a Process Manager I/O, Input/Output Channel (IOC) block from an I/O Processor (IOP).

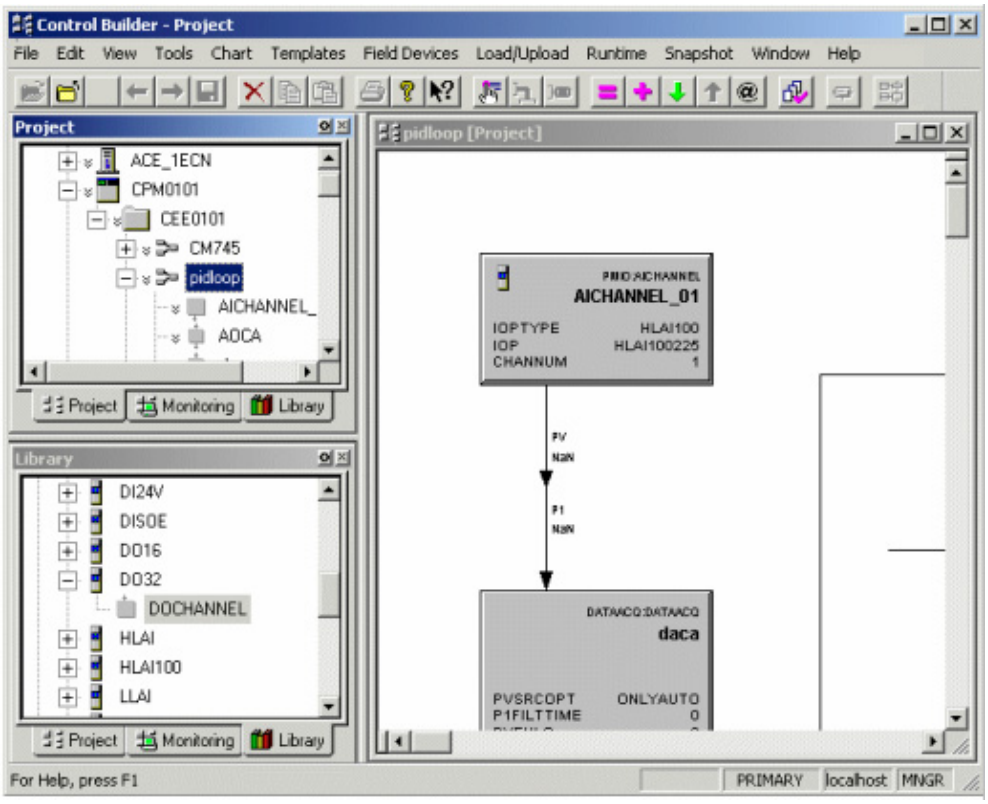
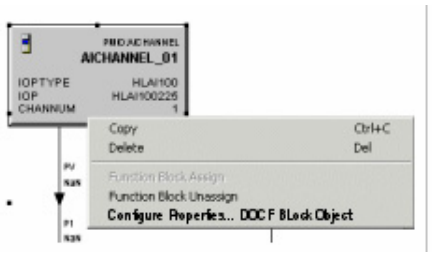
Prerequisites:

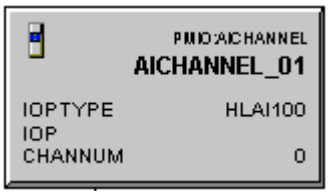
- Control Builder is running
- Tree windows are open

Considerations:

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

Step	Action	Result
1	Double click given CM icon in Project tree.	Opens CM in control chart area.

Step	Action	Result
		
2	Right-click PMIO IOC block to be unassigned.	Opens shortcut menu. 
3	Select Function Block Unassign.	Unassigns IOC block from IOP and associated channel number. This

Step	Action	Result
		changes IOP field to blank and CHANNUM field to 0 on block face. 
4	Close the CM and click Yes.	Closes CM and saves changes.
	ATTENTION You can **not** load a Control Module with an unassigned PMIO IOC block.	
5	See the previous procedure to re-assign the IOC to another IOP.	

Copying a function block

Prerequisites:

- Control Builder is running
- Tree windows are open

Considerations:

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

Perform the steps in the table below to copy an existing function block or function blocks.

ATTENTION

The following connection rules apply when copying function blocks:

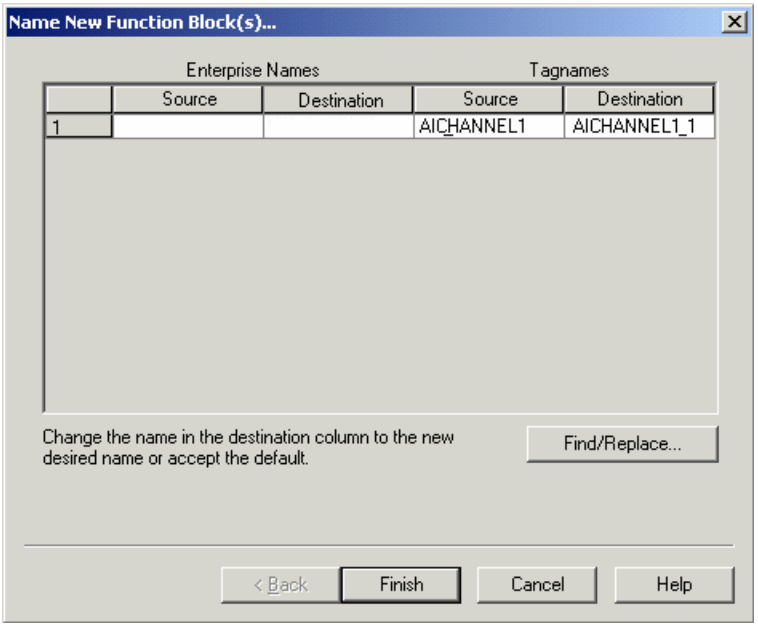
Notes:

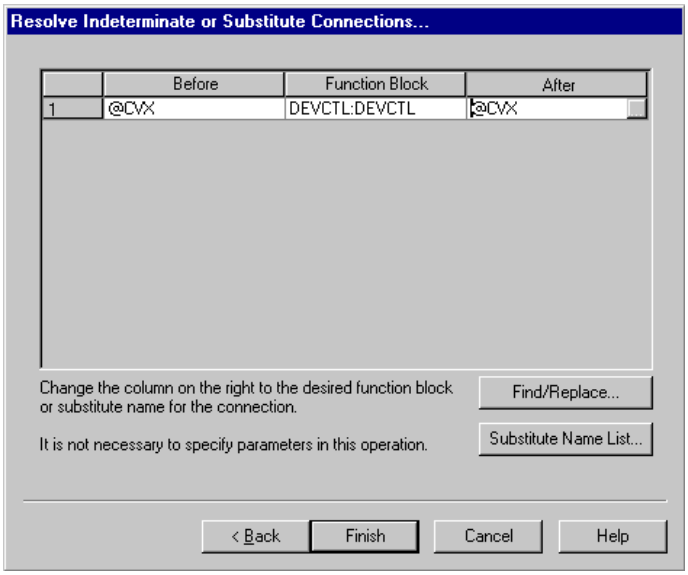
Inside the scope of operation means that the block is included in the list of selected blocks to be copied.

Outside the scope of operation means that the block is not included in the list of selected blocks to be copied.

1. Graphical connections to blocks that are inside the scope of operation are automatically resolved relative to the operation.
2. Graphical connections to blocks that are outside the scope of the operation are not carried over to the new copied blocks.
3. Parameter connections to blocks that are inside the scope of operation are automatically resolved relative to the operation.
4. Parameter connections to blocks that are outside the scope of operation appear in the Connections page to be resolved by the user if desired at the time of copy.
5. Substituted connections on selected blocks always appear in the Connections Page to be resolved by the user if desired at the time of copy.

Step	Action	Result
1	Select the desired function block or block(s) to be copied on the open control drawing.	Selected function block is highlighted on the control drawing.
2	Click Edit -> Copy. Alternate methods: Click <Ctrl>-C. Right-click, then select Copy from the drop-down menu. Tip: Multiple blocks may be copied by shift-clicking and selecting or drawing a selection rectangle (or area) around the blocks.	Selected function block is saved to the Control Builder clipboard.
3	Click Edit -> Paste. Alternate method: Click <Ctrl>-V.	The Name New Function Block(s) dialog appears.

Step	Action	Result
		
4	Change the function block's name in the Destination column of the Name New Function Block(s) dialog to a desired name or accept the default name.	The copied function block is assigned a new name. Note: You may opt to keep the default name which is simply the original name with a number appended to it.
5	Click Next to proceed to the next dialog page (if appropriate) and enter new names as prompted to resolve any existing connections and/or associations.	If the block contains connections to outside blocks, an additional dialog page appears which is used to resolve any existing connections and/or associations (see below).

Step	Action	Result
		
6	Click Finish	The copied block appears on the control drawing.

Moving function blocks within a chart

Prerequisites:

- Control Builder is running
- Tree windows are open

Considerations:

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

Perform the steps below to move function blocks as desired within a chart.

Step	Action	Result
1	Select the desired function block or block(s) on the open control drawing. Note: • Hold down the <Shift> key and click on any additional blocks, as desired, to select multiple blocks to be moved.	Selected function block is highlighted on the control drawing.
2	Hold the left mouse button down and drag (move) the selected block(s) to the desired location.	
3	Release the left mouse button.	Selected block(s) are moved to the desired location.

ATTENTION

A function block will not move if it's destination overlaps another block.

Existing wiring between blocks will automatically resolve itself around the new block layout.

Deleting function blocks

Prerequisites:

- Control Builder is running
- Tree windows are open

Considerations:

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

Perform the steps below to delete function blocks as desired from a particular control strategy.

Step	Action	Result
1	Select the desired function block or block(s) on the open control drawing. Note: • Hold down the <Shift> key and click on any additional blocks, as desired, to select multiple blocks to be moved.	Selected function block is highlighted on the control drawing.
2	Press the <Delete> key.	Selected block(s) are deleted from the Control Drawing.

Using the Parameters Configuration form

The Parameter Configuration Form enables you to customize a function block by changing its name, execution order, and any associated parameters.

Prerequisites:

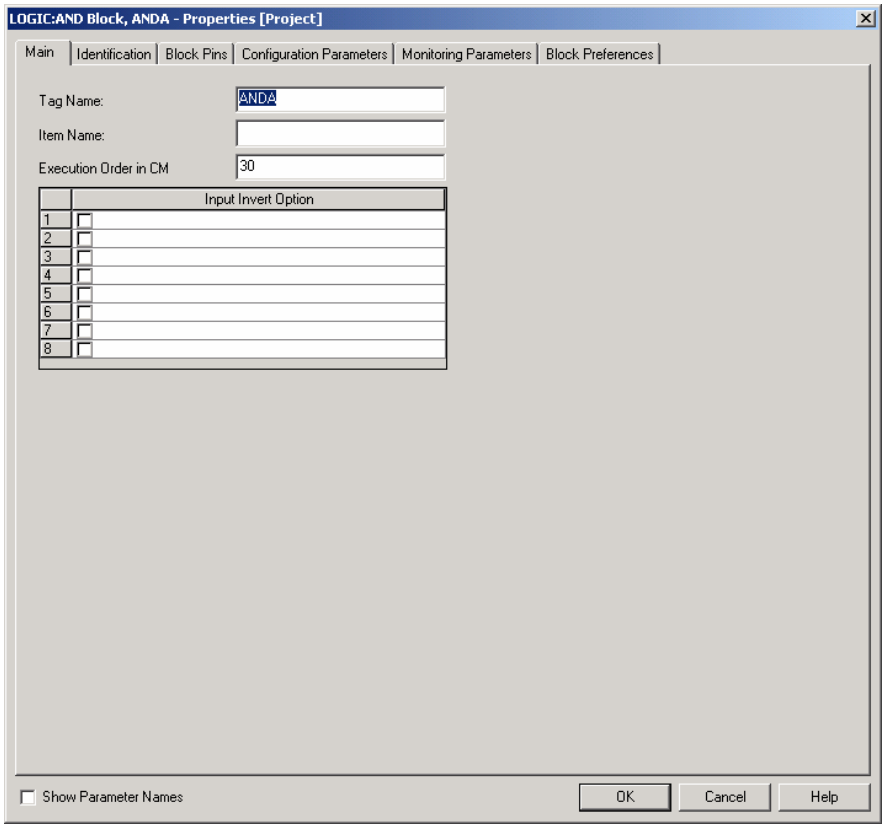
- Control Builder is running
- Tree windows are open

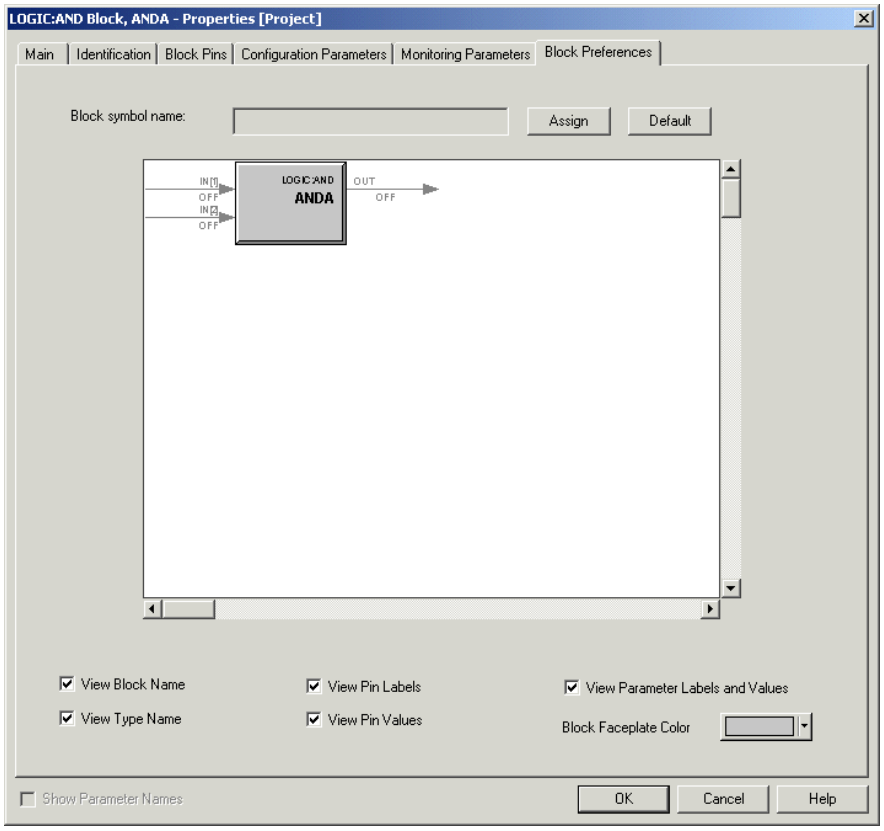
Considerations:

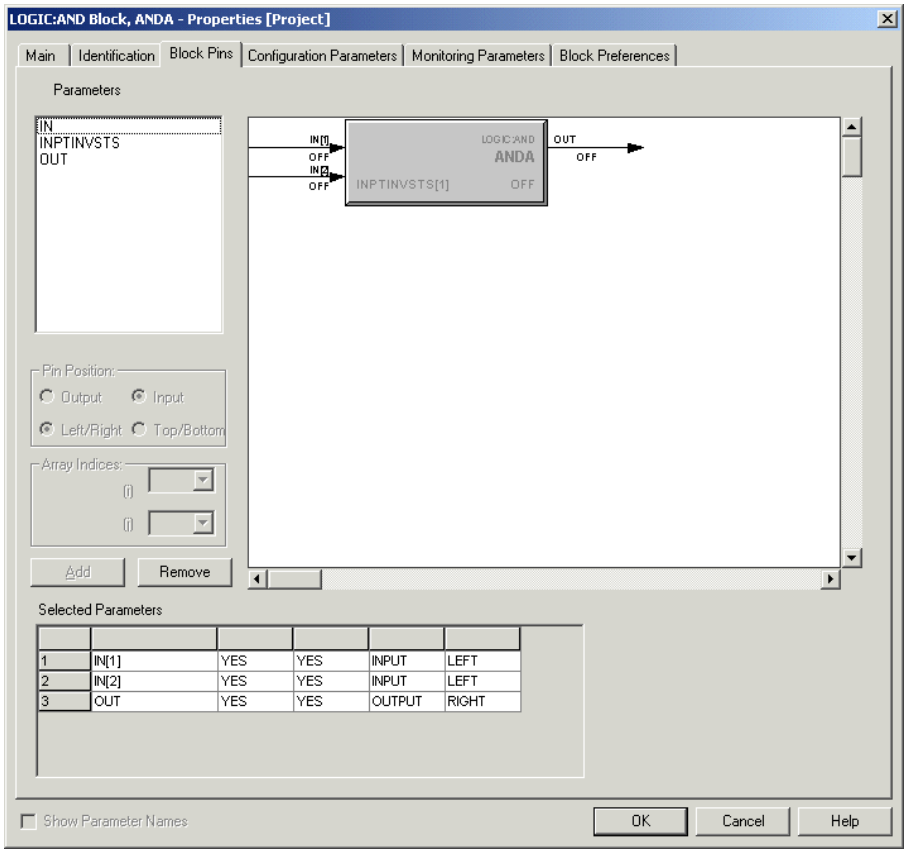
- All illustrations used in the procedure are for example purposes only.
- You have launched Control Builder and created a Control Module that includes an AND function block.
- The tabs that are included with a given block configuration form will vary depending upon block type and active licensed options.
- For parameter configuration help, press the **F1** key to call up help for the active field or click the **Help** button to call up help for the tab.
- The number of configurable parameters for a block varies depending upon whether the block is a data driven block or a custom block. A data driven block requires only a few configured parameters. The AND block is an example of a data driven block. A custom block requires many more configration parameters, including those used specifically for configuring input, output and alarm parameters. The PID-PL block is an example of a custom block.

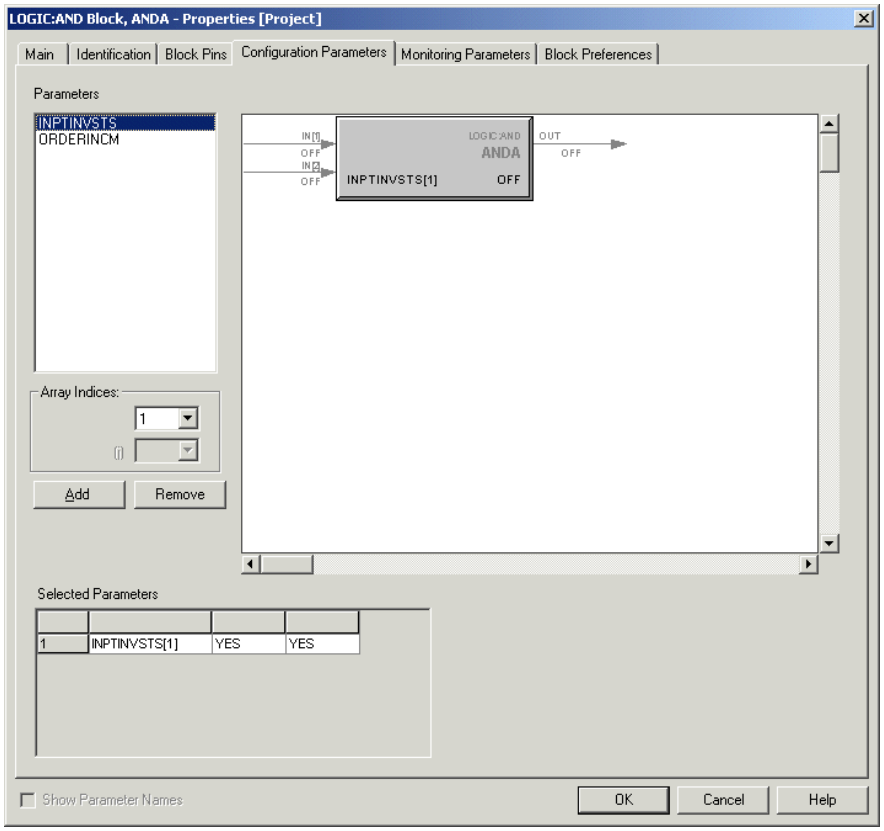
The following procedure outlines the typical steps for configuring an AND block for example purposes and can be easily adapted to apply to other blocks.

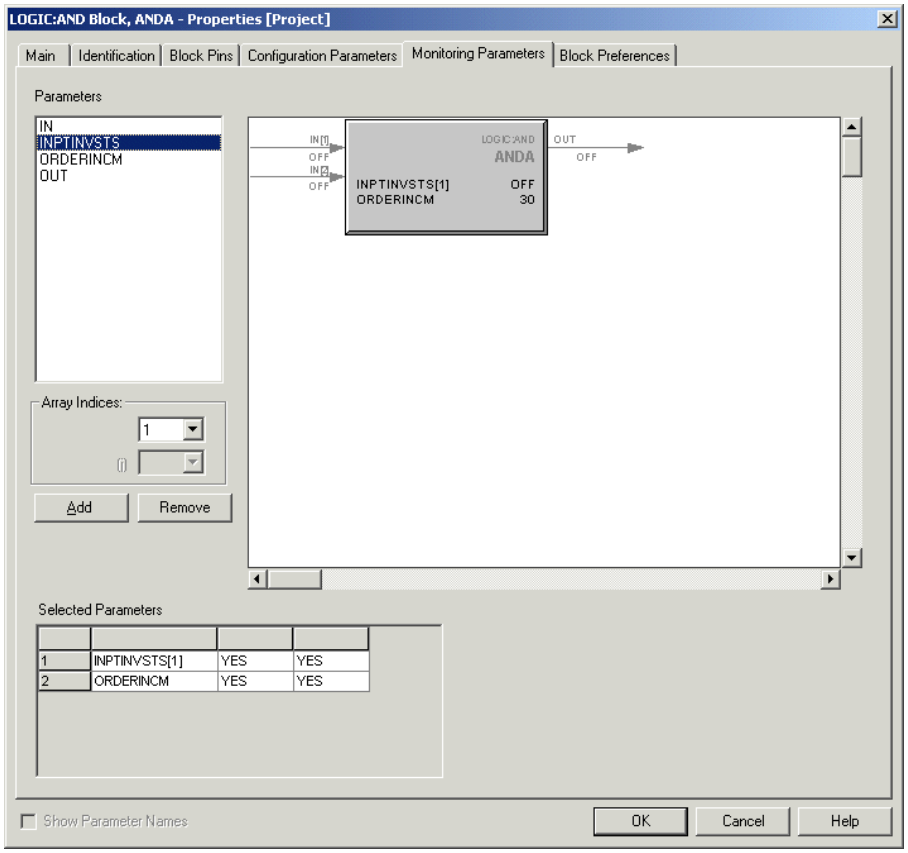
Step	Action
1	On **Project** tab or open Control Module, double-click the **AND** block icon or **AND** block, respectively. Or, right-click the icon or block and select **Configure Parameters . . .** from the list.

Step	Action
	
2	On Main tab, key in desired block name in Tag Name box. Press the Tab key. Press the Tab key.
3	Enter Item Name. Press the Tab key.
4	In **Execution Order in CM** box, key desired value. Press the **Tab** key twice.
5	In row 1of **Input Invert Option** list, click check box to turn option **On** or **Off**. A **check** means the function is **On**. The default is **blank** check box or function is **Off**. Repeat this Step as required.

Step	Action
6	• Click the **Block Preferences** tab. In the view window, check a preview of how the block is displayed in the Control Module control drawing. • Use configuration functions on the tab to change the appearance of the block as desired. Click the **Help** button for more information about the functions. See Appendix E - User Defined Symbols for more information about user definded block symbols.
	
7	• Click the **Block Pins** tab. In the view window, check a preview of the pins currently configured for the block. • Use configuration functions on the tab to change the pin configuration for the block as desired. Click the **Help** button for more information about the functions.

Step	Action
	
8	• Click the **Configuration Parameters** tab. In the view window, check a preview of the parameters currently configured to appear on the block in the **Project** tab. • Use configuration functions on the tab to change the parameter configuration for the block as desired. Click the **Help** button for more information about the functions.

Step	Action
	
9	• Click the **Monitoring Parameters** tab. In the view window, check a preview of the parameters currently configured to appear on the block in the **Monitoring** tab. • Use configuration functions on the tab to change the parameter configuration for the block as desired. Click the **Help** button for more information about the functions.

Step	Action
	
10	Click other tabs to check configuration details, as applicable.
11	Click the **OK** button to close the configuration form and save the changes.
12	This completes the procedure. Repeat the configuration procedure for each function block.

Configuring alarms

Configuration forms for Regulatory Control blocks (such as PID blocks), SCM blocks, Device Control blocks (such as DEVCTL blocks), and various auxiliary blocks (such as DATAACQ blocks) each have an Alarms tab which allows various alarms to be configured.

Prerequisites:

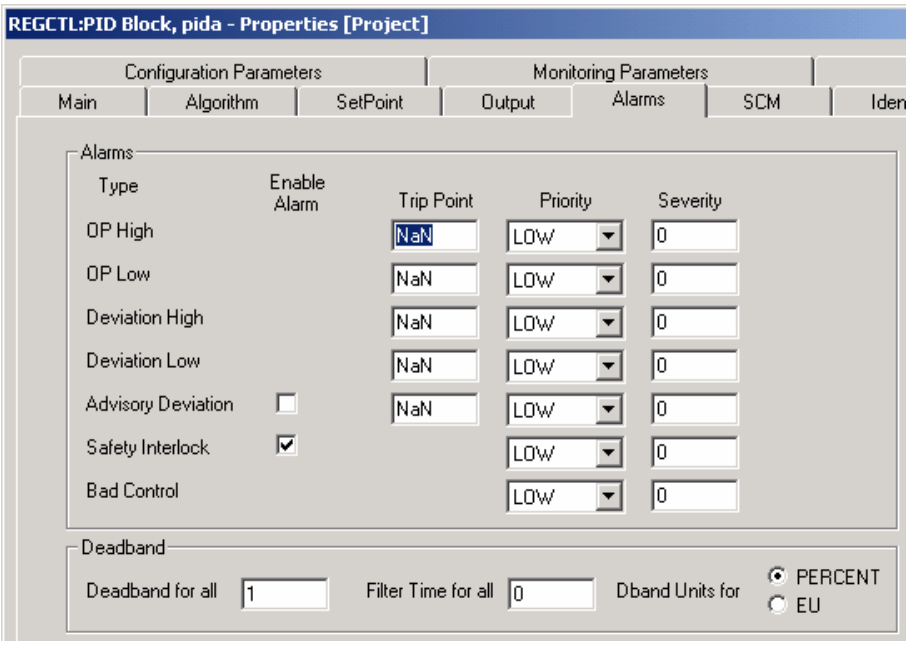
- Control Builder is running
- Tree windows are open

Considerations:

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

Perform the steps in the table below to configure the Alarms tab for a typical block.

Step	Action
1	Double click on the desired function block. The Parameters Configuration Form for the specific function block is displayed. Alternate Method: Right-click on the function block to show the drop-down menu. Click Configure Module Parameters … [Block Name] or [DDCF] Block Object to access the Parameter Configuration form.
2	Select the Alarms tab to access the Alarms form. Note that the Alarms tab for a PID block is used in this example. The Alarms tab for other blocks may include the same or different alarms.
3	Make the necessary changes to the Alarms form (see figure below) and click OK. While in each entry field, press the F1 key to view the Help Topic for the field.



Requesting value changes for configuration parameters

Prerequisites:

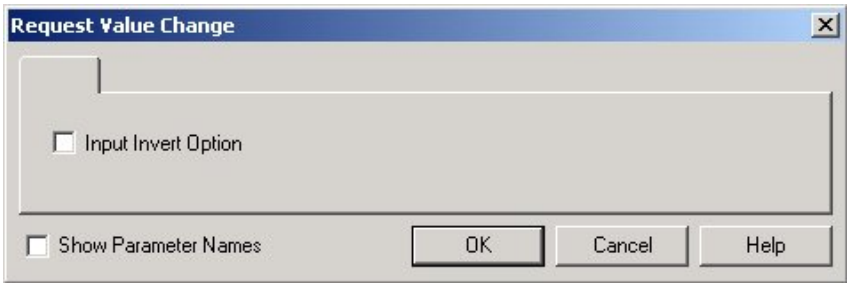
- Control Builder is running
- Tree windows are open

Considerations:

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

Perform the following steps to access parameter values directly from function block symbols in a Control Drawing to request a value change.

Step	Action
1	Double-click on the desired parameter on the function block symbol in the Control Drawing to access the Request Value Change dialog box.

Step	Action
	
2	Enter the new parameter value in the Request Value Change dialog by overwriting the existing value.
	
3	Click OK to enter the new parameter value into the database. Note: These changes are not committed to the database until the chart is saved.

Creating a strategy to use insertion points

This section outlines the tasks involved with creating a strategy for use in an ACE controller that uses a Data Acquistion (DATAACQ) or Regulatory Control block with an insertion point or points from a CAB instance program. In Knowledge Builder, just click the appplicable topic in the following table to jump to the associated reference. The tasks are listed in the suggested order that they should be completed. In most cases, detailed information is provided in other books in Knowledge Builder.



ATTENTION

We recommend that only users who are intimately familiar with the associated function blocks implement control strategies that use insertion points.

You must thoroughly test any control strategy that includes insertion points before using it in an online control system.

Tasks for creating a strategy that uses insertion points
Creating a CAB with insertion program
Creating a control module to include insertion points
Configuring insertion points
Loading control module with insertion points
Activating control module with insertion points
Checking insertion point status
Deleting insertion points

Creating a CAB with insertion program

You create a CAB as you normally would to support your desired insertion program. Refer to

5 Algorithm Block and Custom Data Block User's Guide

You must configure the required parameter references for the block that will be used to call the CAB insertion program. Refer to the Control Builder Component Theory book for more information about insertion points used with

Data Acquisition blocks

or

Regulatory Control blocks

as applicable. Be sure you configure the access level (ACCESSLEVEL) for CAB as continuous control (CONTCONTROL).

Creating a control module to include insertion points

You create a Control Module as you normally would to include the Data Accquisition and/or Regulatory Control blocks that are to include insertion points. You include the CAB instance(s) that are to provide the insertion programs in the same Control Module. Refer to the *Creating a control module* section in this book for details.

Configuring insertion points

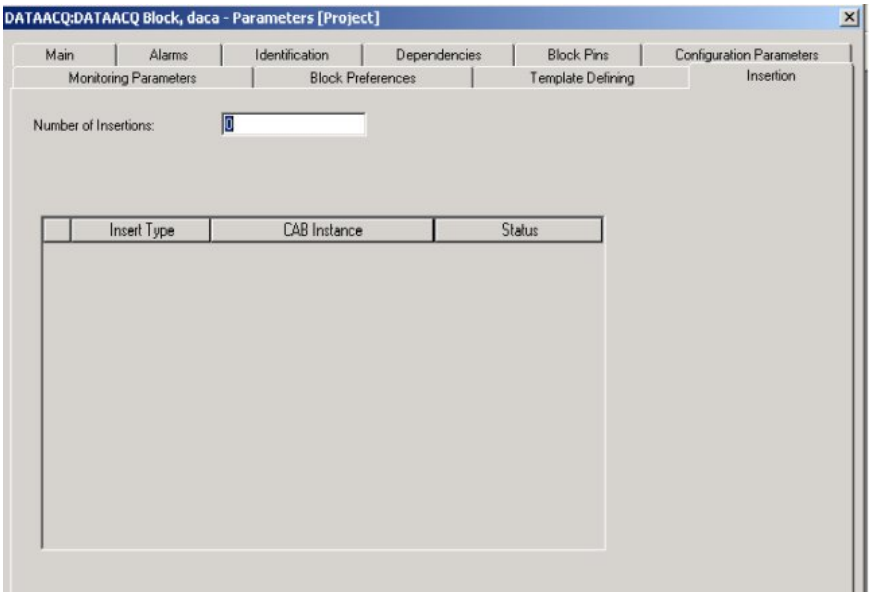
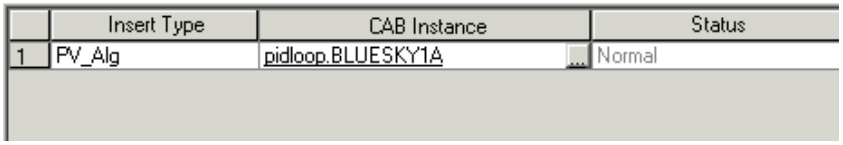
Prerequisites:

- You have created a CAB to provide insertion point support.
- You have created and saved a control module that includes the Data Accquisition and/or Regulatory Control block and the CAB instance to support the associated insertion points.

Considerations:

The following procedure is based on configuring insertion points for a Data Accquistion block. You can easily adapt the procedure to apply for a Regulatory Control block type.

Step	Action
1	In the **Project** tab, locate the Control Module that is to support insertion points, click the plus sign to view its contents, and double-click the Data Accquisition block.
2	On the **DATAACQ Block** configuration form, click the **Insertion** tab.

Step	Action
	
3	Key in **1** in the **Number of Insertions** field and press the **Tab** key to expose row 1 in the table grid.
4	Click in the **Insert Type** column of row 1 and select the type of insertion point from the list.
5	Click in the **CAB Instance** column in row 1. Click the button on the right side of the row in this column, select the CAB instance included in this Control Module in the **Point Selection** dialog, and click the **OK** button to close the dialog and enter the selection in the column row. 
6	Click the **OK** button to close the block's configuration form.
7	In the **Project** tab, double-click the CAB instance to open its configuration form.

Step	Action
8	On the **Main** tab, check the settings for the following parameters. • The **Execution Order in CM** value is identical to the same value for the Data Accquisition block, • The **Access Level** value is CONTCONTROL., and • The **Insertion Point** value is the tag name for the CAB instance.
9	Click the **Parameter References** tab. Click the continue button to the right of any parameter field, select the source of the parameter from the **Point Selection** dialog, click the **OK** button to close the dialog and enter the selection. Repeat this Step as required to idenitfy all needed parameter references.
10	Click the **OK** button to close the configuration form and save the changes.
11	This completes the procedure.

Loading control module with insertion points

You assign and load a Control Module that includes blocks with insertion points to an ACE conntroller the same as you would any Control Module. See the *Loading a control strategy* section in this book for more information.

Activating control module with insertion points

Prerequisites:

- You have thoroughly tested the control strategy with insertion points before using it in an online system.

You activate a Control Module and its contents that include insertion points the same as you would any Control Module. See the *On-line monitoring using Control Builder* section in this book for more information.

Checking insertion point status

Prerequisites:

- You have created and loaded a Control Module that includes blocks with insertion points to an ACE controller.

Step	Action
1	On the **Monitoring** tab, locate the Control Module that includes blocks with insertion points, click the plus sign to view its contents, and double-click the Data Accquisition block or Regulatory Control block which has insertion points configured.
2	On the **DATAACQ Block** configuration form, click the **Insertion** tab.
3	View the **Status** column in the table grid to verify the status of the associated insertion program.
4	Click the **OK** button to close the configuration form.
5	This completes the procedure.

Deleting insertion points

If you delete an insertion point from the **Insertion** Tab for a given block, you should also delete the associated CAB instance from the Control Module. If you want to use the CAB instance as a standalone program, you must first delete the CAB instance, re-configure its ACCESSLEVEL parameter to be PROGRAM through the Parameter Definition Editor, and then restore the CAB instance in the **Project** tab.

Connecting and disconnecting blocks

This section includes information that you can reference for:

Topic	Link
Connecting blocks with insert wire	Click here
Disconnecting blocks	Click here
Repostioning connecting wires using drag and drop	Click here
Repostioning connecting wires using vertices	Click here
Connecting blocks with Parameter Connector option	Click here
Cross-references function	Click here
Enabling cross-references	Click here
Printing cross-references	Click here
Using Point selection tool	Click here
Using peer-to-peer communincations	Click here

Connecting blocks with insert wire

A control strategy is created by connecting function blocks to each another.

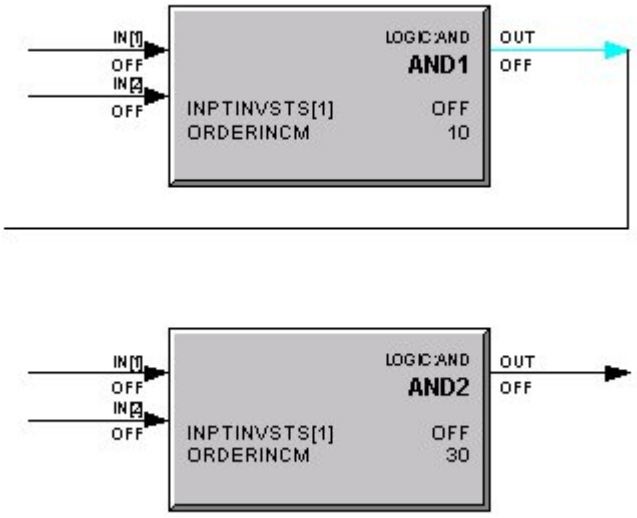
Prerequisites:

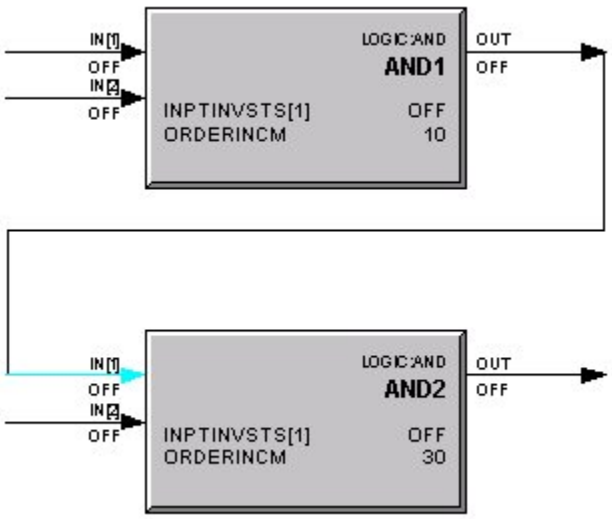
- Control Builder is running
- Tree windows are open

Considerations:

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

The following table describes how to connect function blocks using the Insert Wire option.

Step	Action
1	Click Insert -> Wire and select the desired pin. Alternate methods: • Click on the Wire toolbar button  . • Double-click on the desired pin. The cursor changes to a cross-hair.
2	Continue clicking along the desired wire route. Each click completes the path to that point. However, you can click once on the destination block pin, and let Control Builder route the wire automatically for you. 

Step	Action
3	Click on the final connection point. In the example below, five mouse clicks were made.  The pins are connected with a solid wire.

ATTENTION

Wires may automatically resolve to a more direct route when blocks are moved within the Control Module.

REFERENCE

- Press the ESC key to clear all wire entries if the final connection has not yet been made.
- To wire blocks together in a hurry, double-click on the desired parameter pin in the first block so that the cursor changes to a “+”, then click the desired parameter pin on the destination block so that a wire appears.

Disconnecting blocks

Prerequisites:

- Control Builder is running
- Tree windows are open

Considerations:

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

Step	Action	Result
1	Click on the wire to be deleted.	Connecting wire is selected.
2	Click Edit -> Delete, or press the Delete key.	The wire disappears, the connection is broken.

Repositioning connecting wires using drag and drop

Prerequisites:

- Control Builder is running
- Tree windows are open

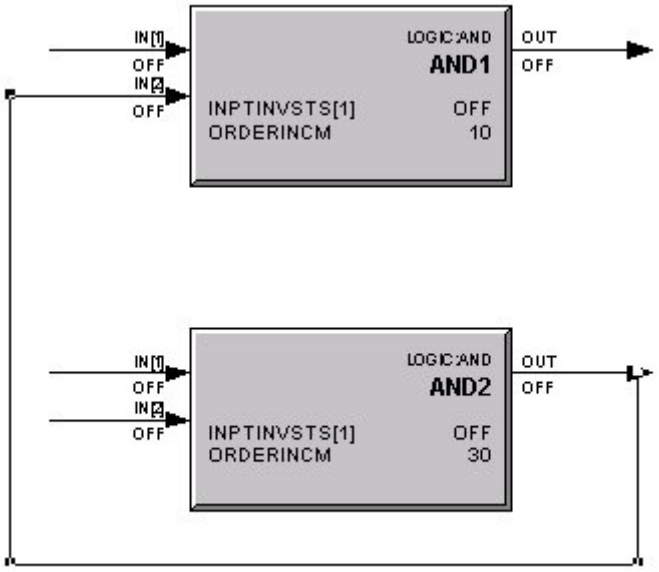
Considerations:

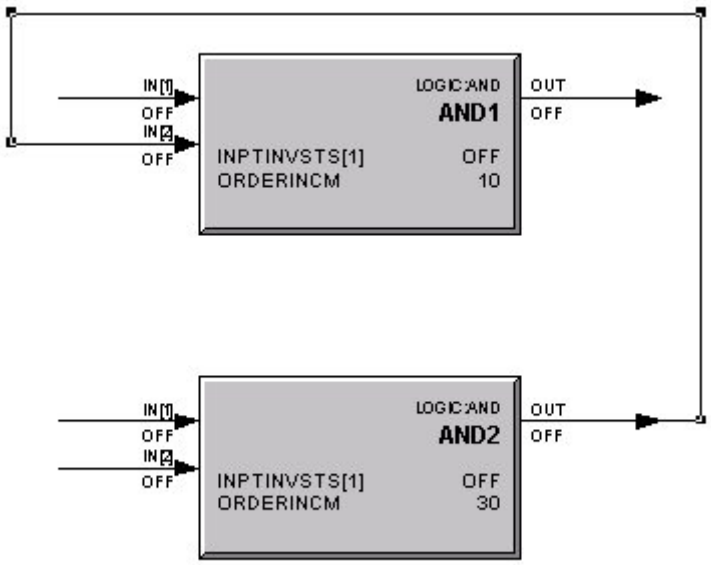
- All illustrations used in the procedure are for example purposes only.

ATTENTION

Wires may automatically resolve to a more direct route when blocks are moved within the Control Module.

Follow the steps in the table below to reposition an existing wire.

Step	Action
1	Click on the wire to be repositioned.  Connecting wire between AND1.IN[2] and AND2.[OUT] is selected.

Step	Action
2	Drag the desired vertices to the new position and release. The wire snaps to the new position.  The lower-left corner wire vertex has been positioned above the Block Symbols in the upper-right corner.

Repositioning connecting wires using vertices

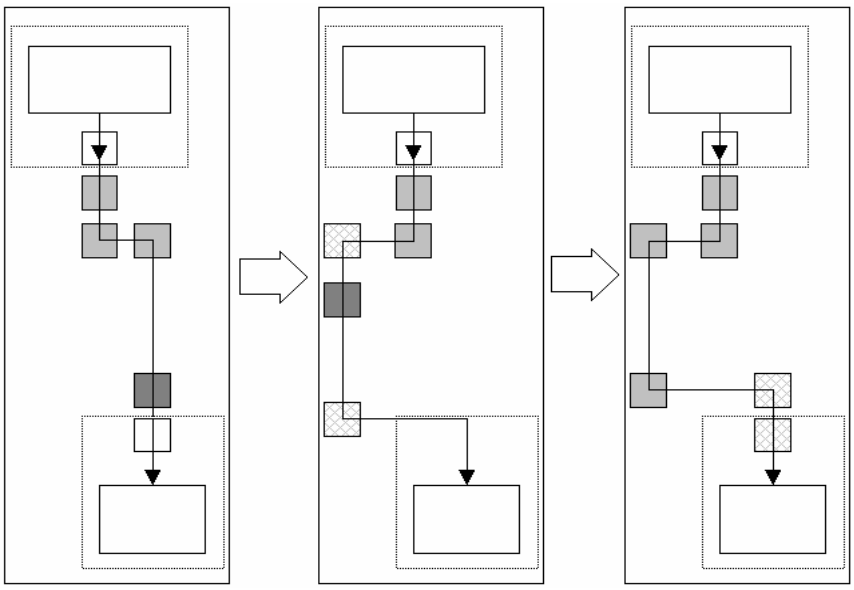
Prerequisites:

- Control Builder is running
- Tree windows are open

Considerations:

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

Step	Action
1	Click once on a vertex, the dark area at the corner of a wire. In the left illustration, this vertex is the dark square.
2	Drag the vertex to a new location. In the middle illustration, the vertex has been moved to the left, creating two new vertices.
3	Release the vertex. The original vertex will resolve itself, leaving the new vertices, as shown in the right illustration. A maximum of 50 vertices may be present in a wire. To remove vertices, drag the wire into a straight line and the middle vertices will resolve themselves.



Connecting blocks with Parameter Connector option

To connect function blocks within different Control Modules, the Insert Parameter Connector option is used. This option may also be used to connect blocks within the same Control Module, if it is difficult to route a connecting wire.

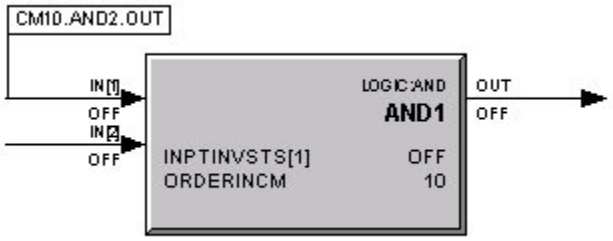
Prerequisites:

- Control Builder is running
- Tree windows are open

Considerations:

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

Step	Action	Result
1	Click the Parameter Connector toolbar button  . Alternate method: Click Insert -> Parameter Connector.	The pointer turns into a cross-hair.
2	Click on the desired input block pin.	
3	Double-click outside the block at the required Parameter Connector location.	Parameter Connector appears. Tip: Clicking in different places outside the block symbol results in the Parameter Connector being placed in different locations. Try single-clicking to establish a path to a desired location before double-clicking. Press the ESC key to cancel the operation and start over if desired.

Step	Action	Result
4	Type in the full name of the desired block connection, including Control Module.Block Name(Tag).Parameter and press the <Enter> key. Alternate method: Click on the button with the dot leader icon (three dots) to access the Point Selection dialog. Use this dialog to find the desired point name and parameter and then click on the button titled Select to insert the specified parameter into the parameter connection field. Then click the Close button on the top right-hand corner of the Point Selection dialog to close the dialog and return to the control drawing.  In this example, the OUT parameter of the block AND2 on Control Module 10 is entered.  This example illustrates a completed parameter connection.

ATTENTION

- The full Control Module.Block Name(Tag).Parameter name must be entered, even if the two blocks to be connected are in the same Control Module.
- If an invalid parameter is entered, you will be prompted by a pop-up dialog.
- Refer to Creating a Substitute Name List for information on creating a substitute name which can be used when connecting blocks using the parameter connector option.

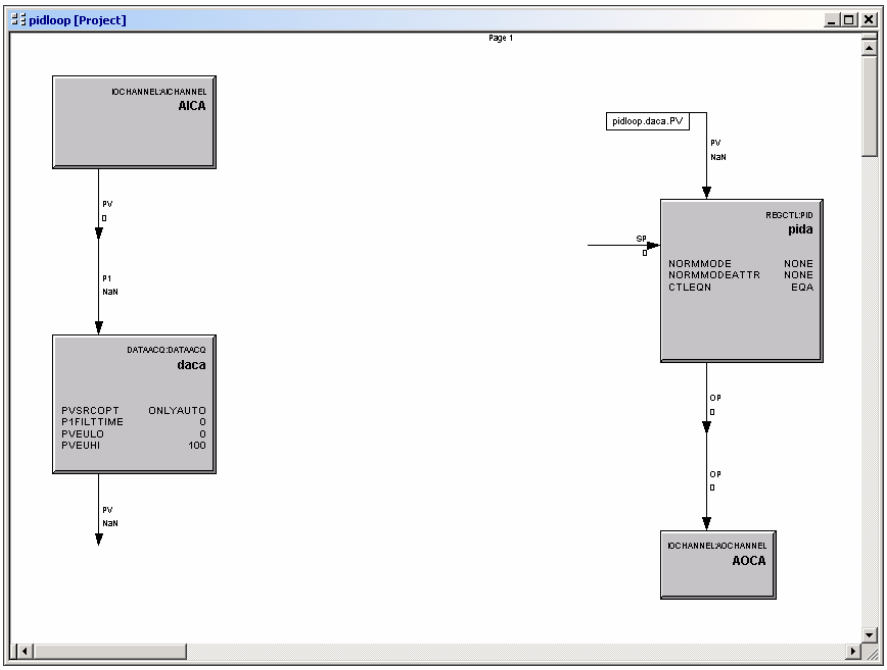
Cross references function

Cross-References are used to visually identify the source of a Parameter Connector in Control Builder. It will appear as a box alongside the Block with the Block name and full pin name.

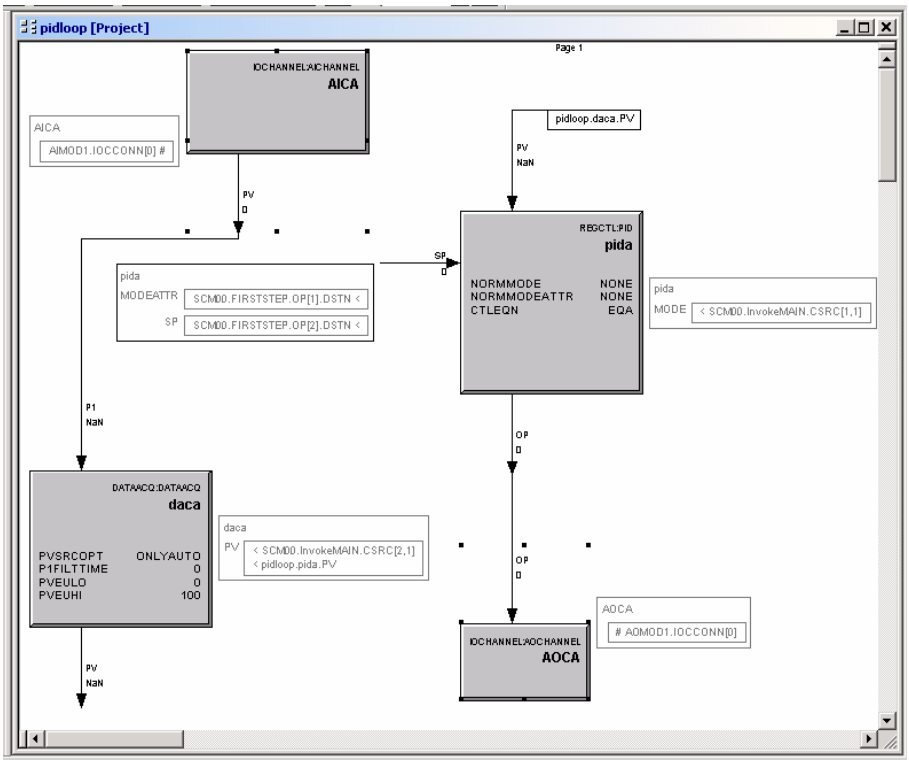
Cross References are dynamically created as a by-product of:

- a Parameter Connector
- an Expression in an SCM Block or a CM's Auxiliary Block
- an Assigned Input or Output Block

Input Cross References will have a "#" symbol before the full pin name. Output Cross References also will have a "#" symbol, but after the full pin name as illustrated in the "enabled" figure below. The "<" symbol is used for Expressions and all other cases.



Parameter connector with cross-references disabled



Parameter connector with cross-references enabled

Enabling cross-references

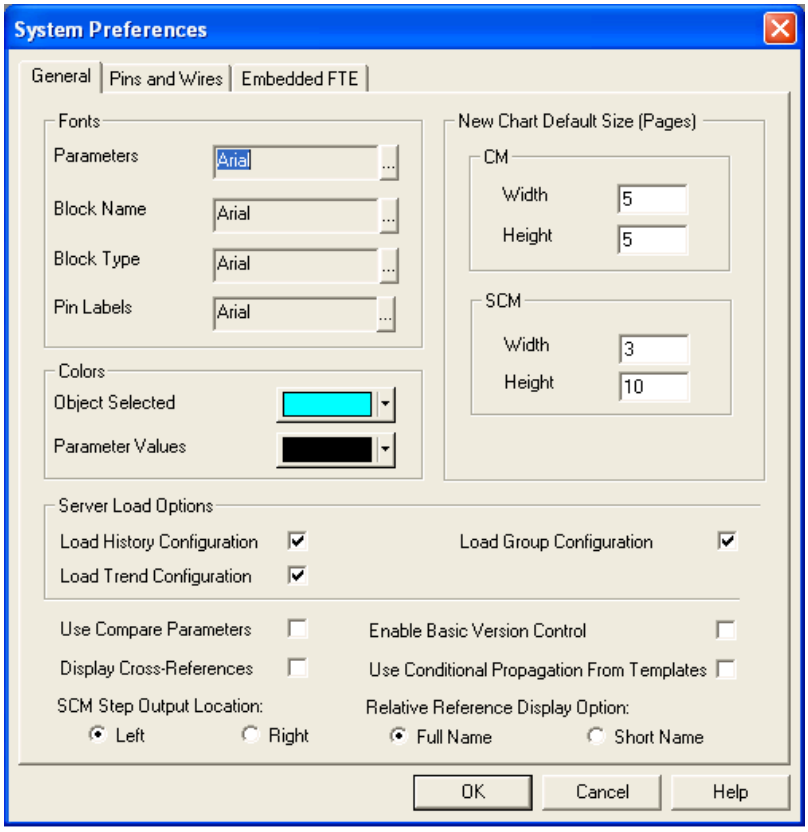
Prerequisites:

- Control Builder is running
- Tree windows are open

Considerations:

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

Step	Action
1	Start Control Builder. Click Tools -> System Preferences to open the dialog.



Step	Action
2	On the General tab, verify the Display Cross-References check box is **checked** or enabled. Click the check box, if it is **blank** or disabled.
3	Click OK to close the System Preferences dialog.

Printing cross-references

Cross-References will appear in hardcopy charts printed from either Project or Monitor mode.

Using the Point Selection tool

Prerequisites:

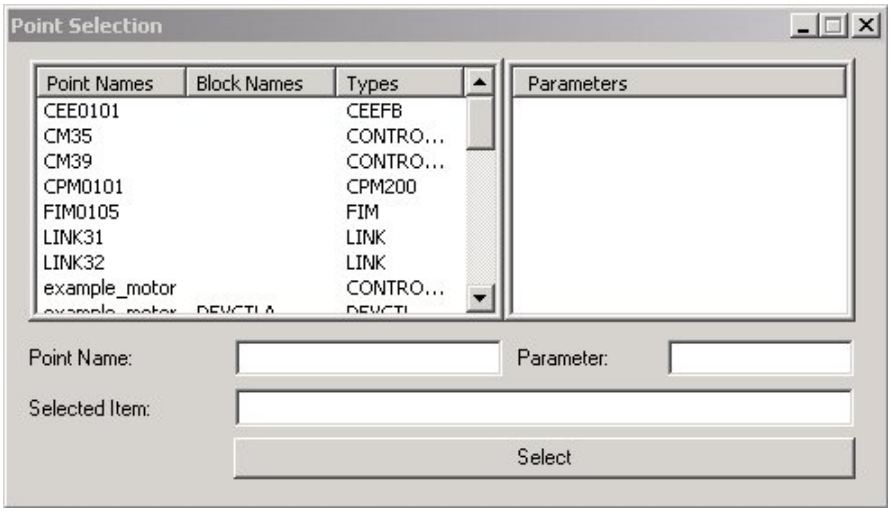
- Control Builder is running
- Tree windows are open

Considerations:

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

Perform the following steps to use the Point Selection tool to find a desired point name and parameter when referencing a particular parameter expression.

Step	Action	Result
1	Click Tools -> Point Selection to access the Point Selection dialog. Alternate methods: • Click the Point Selection toolbar button  . • Click on the Ellipsis button  associated with a Parameter Connection or Peer-to-Peer Connection. • Select the Points button  on an Expression tab, such as those associated with AUXCALC, REGCALC, STEP and TRANSITION blocks.	Point Selection dialog appears on the screen (see below).



Step	Action	Result
2	Select the desired Point Name from the list of Point Names and associated Block Names/Types on the Point Selection dialog. Tip: • Click the appropriate column heading (Point Name, Block Name, or Block Type) to sort the column's list alphabetically in the left pane of the Point Selection dialog. Click the column heading twice to arrange the column's list in reverse alphabetical order.	Point Name is highlighted in the list and a list of related parameters appears in the right-hand listbox on the Point Selection dialog. Selected Point name appears in the Point Name field and Selected Item field on the lower portion of the Point Selection dialog.
3	Select the desired parameter from the list of parameters in the right-hand listbox on the Point Selection dialog.	Selected parameter is highlighted. Selected parameter appears in the Parameter field and as part of the Point Name in the Selected Item field on the lower portion of the Point Selection dialog.
4	Click the Select button.	Selected point (Point Name plus specified parameter name) appears as a tagname in the corresponding parameter expresssion. Note: An error expression appears if you attempt to enter a parameter that is not appropriate for the designated parameter expression.
5	Click the Close button on the top right-hand corner of the Point Selection dialog to close the dialog and return to the control drawing.	

Using peer-to-peer communications

Peer-to-peer communications is a function in Control Builder whereby data can be shared among separate tagged blocks irrespective of CEE location. That is, if a connection can legitimately be formed between two blocks within the same CEE, then that connection can be formed between two of the same type of blocks which are in separate CEEs.

Prerequisites:

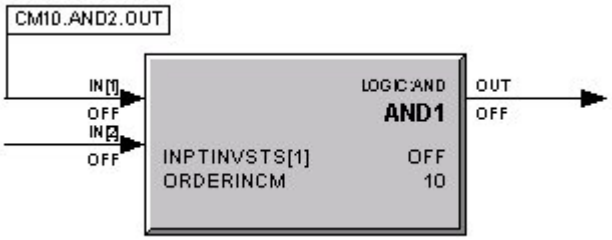
- Control Builder is running
- Tree windows are open

Considerations:

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

Perform the following procedure to establish peer-to-peer communications:

Step	Action
1	Designate (determine) the definition (def) block in the subscribing CEE. • The definition block defines the parameter value which would be useful for the reference block to acquire or possibly modify. • Refer to the Experion Control Builder Components Theory, Peer-to-Peer Functionality, Basic Peer-to-Peer Design Concepts and Implications for Control Builder Configuration for rules and tips for configuring the def block.
2	Designate (determine) the reference (ref) block in the subscriber CEE. • The reference block receives parameter data from the definition block so long as the data type of the reference parameter matches the data type of the definition parameter. • Refer to the Experion Control Builder Components Theory, Peer-to-Peer Functionality Basic Peer-to-Peer Design Concepts and Implications for Control Builder Configuration for rules and tips for configuring the ref block.
3	Right-click on the subscribing CPM's icon symbol in the Project tab, then click Configure Module Parameters. Set the update rate for the subscribing CEE to a desired value. Click OK to close the configuration form. • The update rate is the rate at which the publishing CEE publishes its data to its subscriber. The update rate is set for the whole of the CEE and hence all references are published at that rate. Refer to the Experion Control

Step	Action
	Builder Components Theory, Peer-to-Peer Functionality, Basic Peer-to-Peer Design Concepts and Implications for Control Builder Configuration for the maximum number of parameter references allowed for each update rate.
4	Click on Insert -> Parameter Connector. The chart containing the reference block must have the focus. The pointer turns into a cross-hair.
5	Click on the desired input pin on the reference block.
6	Double-click at the required Parameter Connector location in the same CM. Tip: Clicking in different places outside the block symbol results in the Parameter Connector being placed in different locations. Try single-clicking to establish a path to a desired location before double-clicking. Press the ESC key to cancel the operation and start over if desired.
7	Type in the full name of the desired definition block parameter, including Control Module.Block Name(Tag).Parameter and press the <Enter> key. Alternate method: Click on the button with the dot leader icon (three dots) to access the Point Selection dialog. Use this dialog to find the desired point name and parameter and then click on the button titled Select to insert the specified parameter into the parameter connection field. Then click the Close button on the top right-hand corner of the Point Selection dialog to close the dialog and return to the control drawing.  In this example, the OUT parameter of the block AND2 on Control Module 10 is entered. 

Step	Action
	This example illustrates a completed parameter connection.

ATTENTION

- This example uses a “named” connector which is a parameter connector configured as an input to a reference block. A “named” connector could also be a named parameter within an expression (such as an SCM Step output).
- If an invalid parameter is entered, you will be prompted by a pop-up dialog.

Associating I/O Channels to I/O Modules (IOMs)

Each I/O that you have identified must be associated with an appropriate analog I/O block or digital I/O block. This association gives the physical I/O a logical connection to the control strategy.

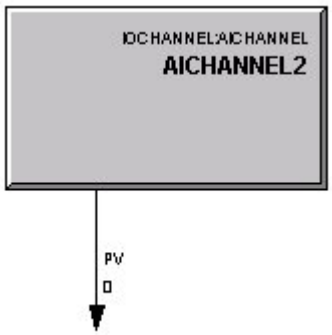
Prerequisites:

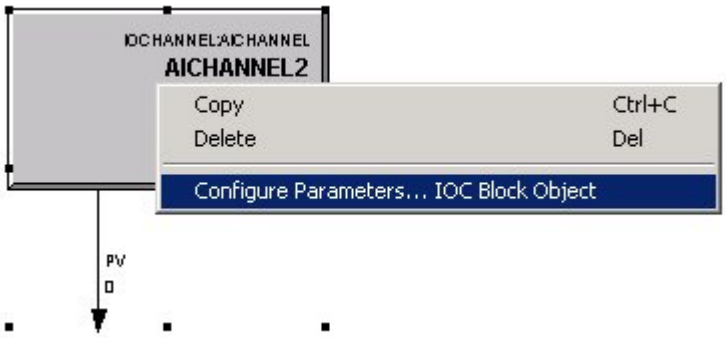
- Control Builder is running
- Tree windows are open

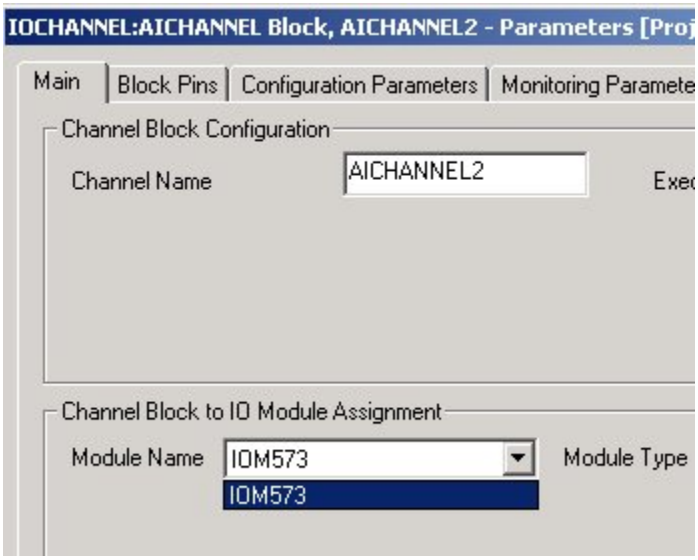
Considerations:

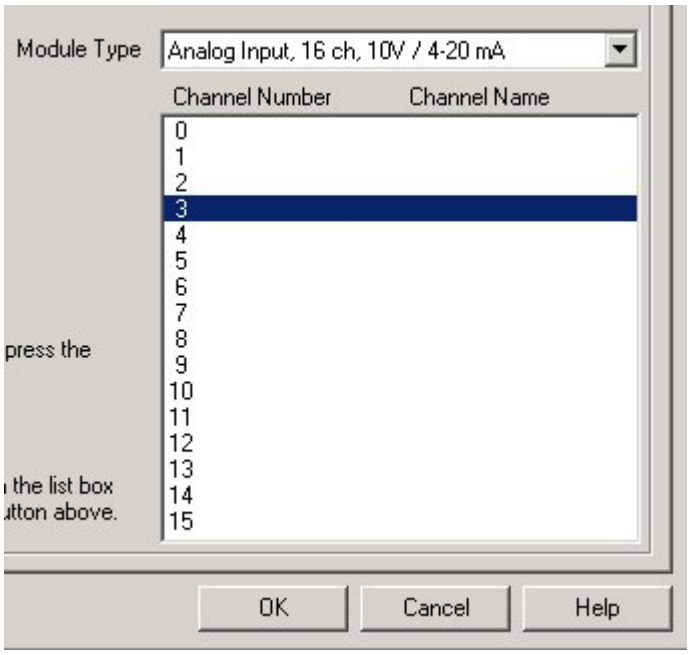
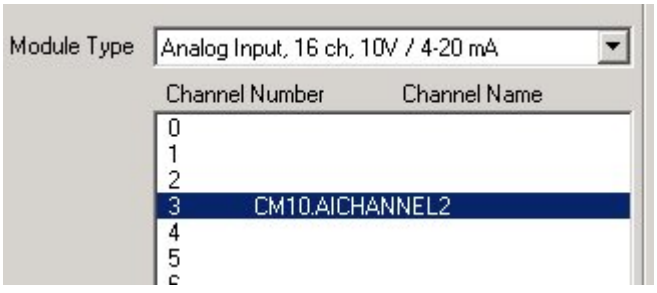
- All illustrations used in the procedure are for example purposes only.

Follow the steps in the table below to associate chassis I/O channels (IOC) to I/O Modules.

Step	Action
1	Right-click on the desired IO Channel block. 

Step	Action
2	Click on Configure Module Parameters … IOC Block Object.  or, double-click on the desired IO Channel block.

Step	Action
3	On the Parameters Configuration form (Main tab), click on the Module Name drop-down box and select the I/O Module to which this I/O channel block is to be associated.  In the above example, AICHANNEL2 will be associated with IOM573.

Step	Action
4	Click on the channel number within this I/O module that is to contain this I/O channel block. 
5	Click on the Assign Channel Block button. The I/O channel block has been assigned to channel number 3 in the I/O module. 

REFERENCE

To Unassign an I/O channel block, click on the I/O channel block on the Channel Parameter Configuration form in the Channel Block to IO Module Assignment portion of the window and click on the Unassign Channel Block button.

Inserting OLE objects into charts

Object linking and embedding (OLE) is a feature supported by Control Builder which enables you to transfer and share information between a third-party Windows-based application and a control chart. Typically, you might want to insert a Microsoft Word (or Wordpad/Notepad/Clipart) file to add an annotation to a chart. Various file formats are available depending on what third-party applications are installed on your computer. Refer to the following procedures and the separate third-party application documentation for complete instructions.

Prerequisites:

- Control Builder is running
- Tree windows are open

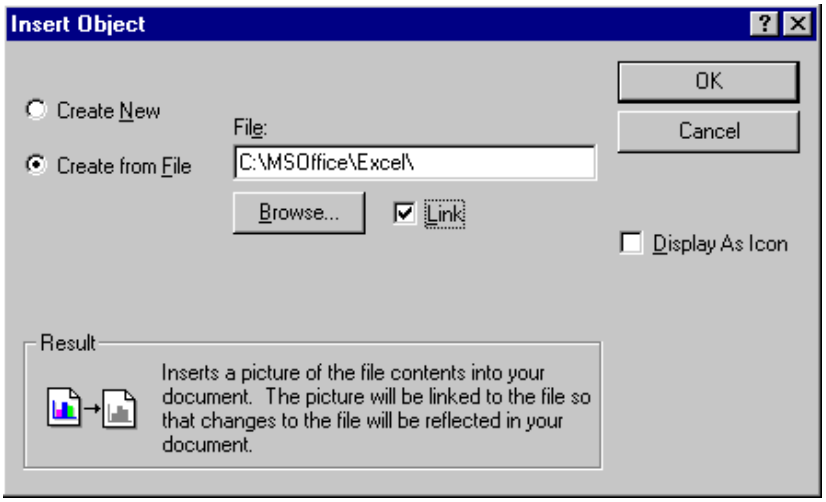
Considerations:

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

Linking an OLE object into a chart

Perform the following steps to link an OLE object into a chart.

Step	Action	Result
1	With the chart open on the screen, move the insertion point to the place where you want the object inserted.	
2	Click Insert -> OLE Object.	An Insert Object dialog box appears on the screen (seebelow).

Step	Action	Result
		
3	Select either the Create New or Create from File radio button as appropriate: • Select Create New to create a new file. • Select Create from File to use an existing file and specify the proper pathname in the File entry field. If necessary, click on the Browse button to help locate the desired file.	
4	Check the Link checkbox to indicate this is a linked file.	
5	Click OK in the Insert Object dialog box.	Linked file icon appears in the upper left-hand corner of the chart..

Editing a linked file in a chart

Prerequisites:

- Control Builder is running
- Tree windows are open

Perform the following steps to edit a linked file in a chart.



ATTENTION

When you link an object, you are not making a copy of the information, you are creating a reference, or link, to the document that contains the information (source document). When you edit a linked object, you are actually editing the information in the source document. The destination document (that is, the chart) only contains a link to where the object exists in the source document.

Step	Action	Result
1	Double-click on the icon that represents the linked file.	The linked file opens in the corresponding application.
2	Make any desired changes to the file using the separate third-party application.	
3	Save the changes and close the third-party application.	The edited file is placed on the chart.

Embedding an OLE object into a chart

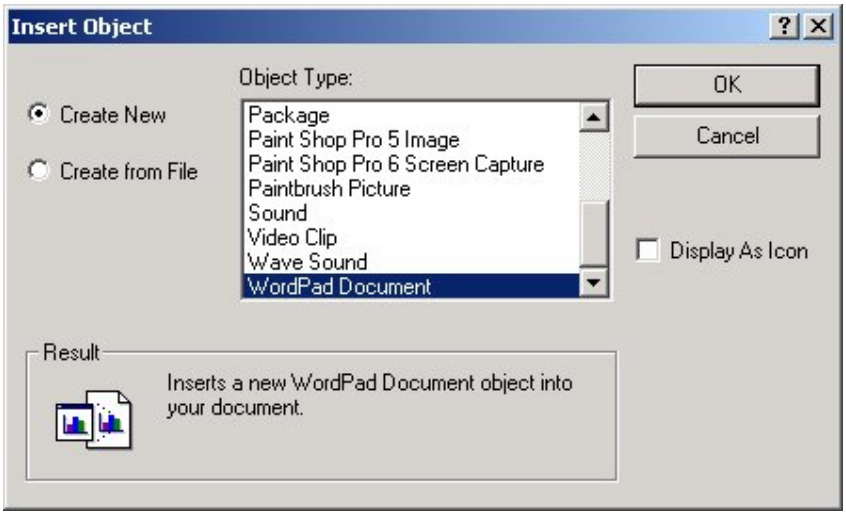
Prerequisites:

- Control Builder is running
- Tree windows are open

Considerations:

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

Perform the following steps to embed an OLE object into a chart.

Step	Action	Result
1	With the chart open on the screen, move the insertion point to the place where you want the object inserted.	
2	Click Insert -> OLE Object.	An Insert Object dialog box appears on the screen.
3	Select either the Create New or Create from File radio button as appropriate: • Select Create New to create a new file. • Select Create from File to use an existing file and specify the proper pathname in the File entry field. If necessary, click on the Browse button to help locate the desired file.	All applications on your computer that support object linking and embedding are listed.
		
4	Select the desired third-party application from the list of applications and click OK.	The desired application opens.
5	Use the selected application to create the object to be embedded in the chart.	Desired object is embedded in the chart.

Editing an embedded file in a chart

Prerequisites:

- Control Builder is running
- Tree windows are open

Perform the following steps to edit an embedded file in a chart.



ATTENTION

When you embed an object, you make a copy of the object in the source document and transfer this copy to the destination document (that is, the chart). You no longer have any connection to the document from which you transferred the information. When you edit an embedded object, the source document is not affected. For example, suppose you embed a drawing into a chart. If you change the embedded drawing from inside the chart, the drawing in the source document is not affected.

Step	Action	Result
1	Double-click on the object in the chart.	The appropriate application opens on the screen.
2	Make the desired edits using the corresponding third-party application.	Desired edits are made.
3	Save the edited object and close the third-party application to return to the Control Builder chart.	

Deleting an OLE object from a chart

Prerequisites:

- Control Builder is running
- Tree windows are open

Perform the following steps to delete an OLE object from a chart.

Step	Action	Result
1	Select the object.	Object is selected on the chart.
2	Either select Edit -> Delete or click the <Delete> key.	Object is deleted from the chart.

Using Server Scripting in Control Builder

This section includes information that you can reference for:

Topic	Link
Server Sripting overview	Click here
Using Script Editor	Click here

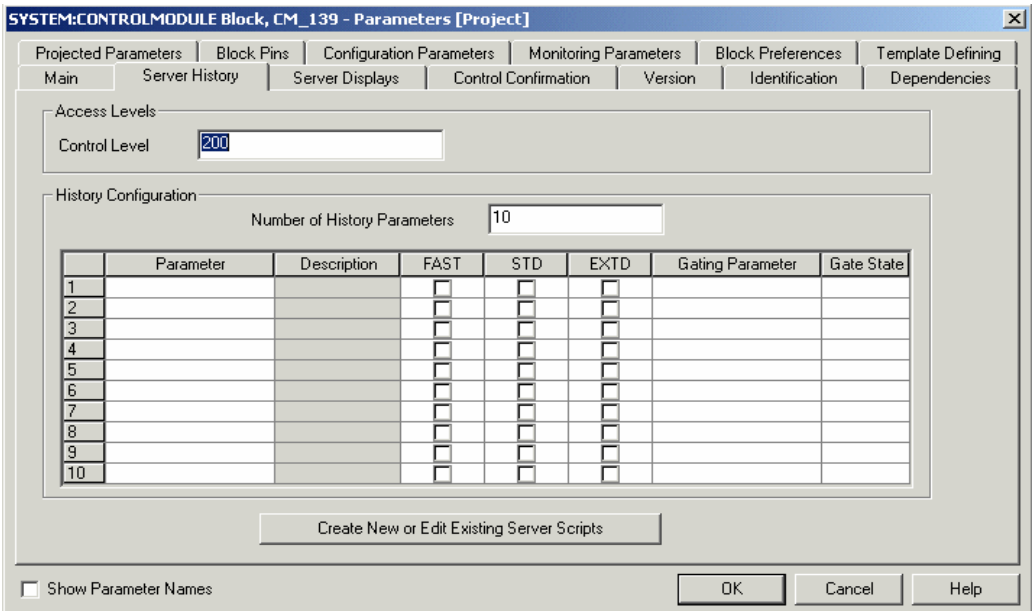
Server Scripting overview

The Server Scripting feature allows scripts to be attached to various objects within the Experion system, including points and parameters. These scripts are run whenever certain trigger events occur, such as a parameter change or a point entering or leaving an alarm state.

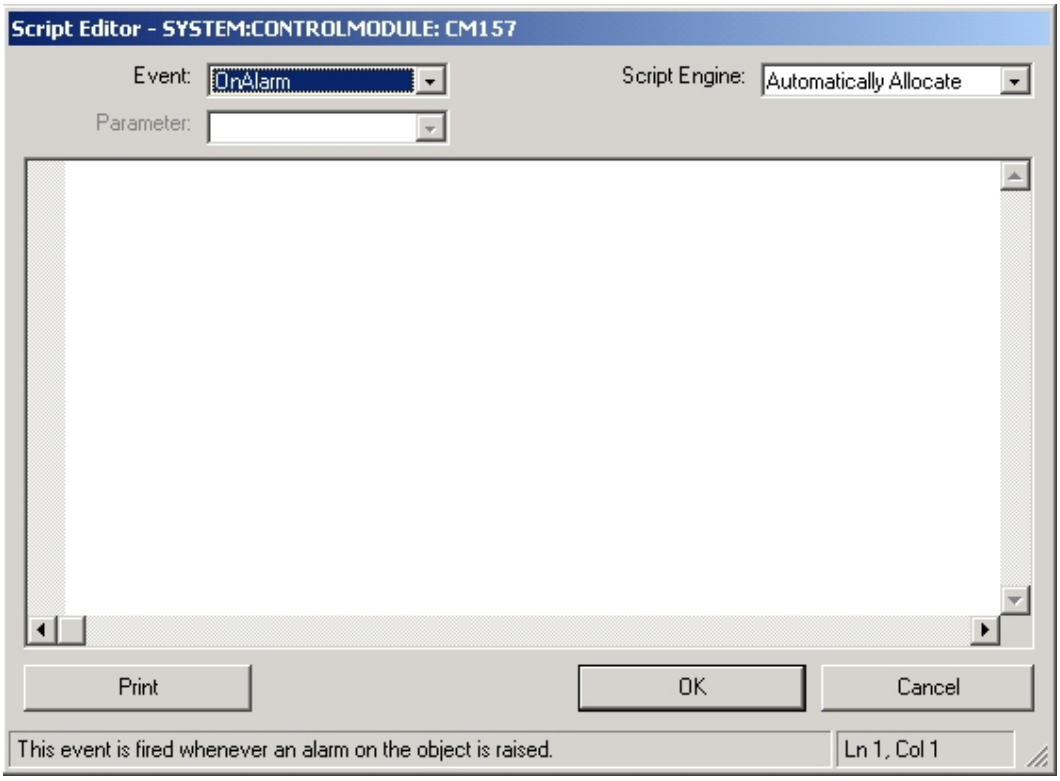
To create or write a script in Control Builder, use the instructions below to access the Script Editor. For more detailed instructions on Server Scripting functions and creating scripts, refer to the *Server Scripting Reference* in Knowledge Builder.

Using Script Editor

Open the CM properties screen and select the Server History tab.

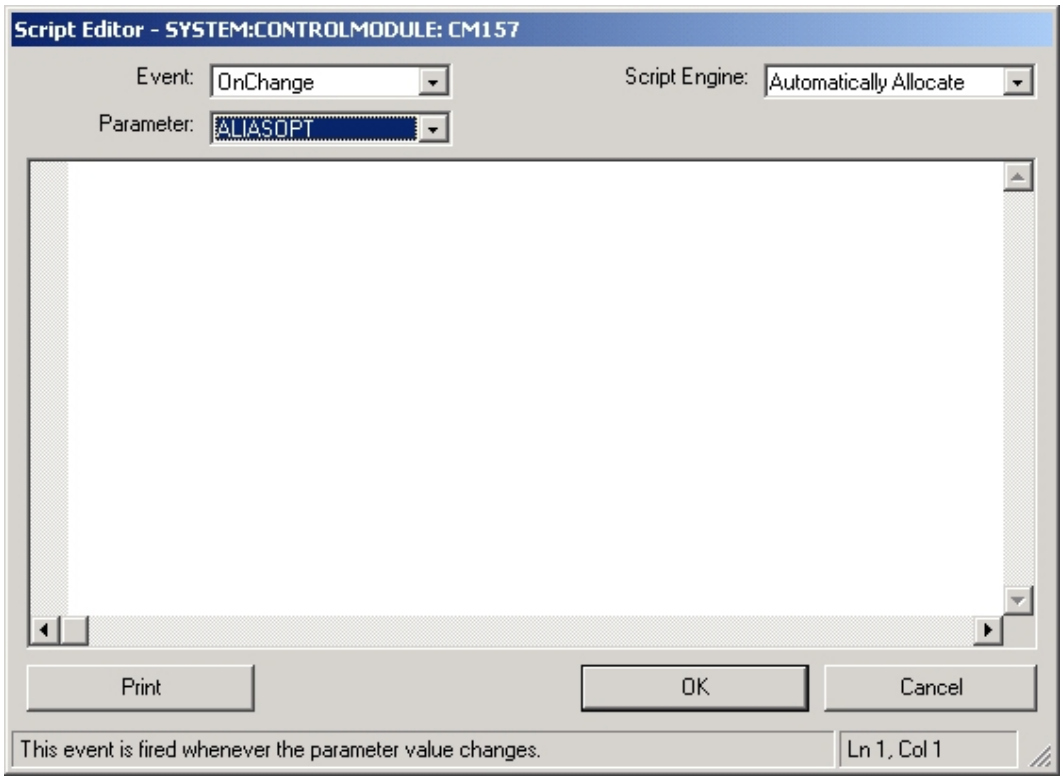


When the “Create New or Edit Existing Server Scripts” button is selected.. The screen allow the editing of all scripts for a particular point, and all scripts for parameters of that point.



Script Editor for a point

If OnChange or OnOperChange are selected as the Event type, all the parameters of the container block and the basic blocks contained by the container block will be listed in the Parameter combo box, and any parameter can be tied to the script.



Script Editor for a point parameter

For detailed information on using Server Scripting, refer to the *Server Scripting Reference* in Knowledge Builder.

Printing from Control Builder

This section includes information that you can reference for:

Topic	Link
Printing features	Click here
Printing preferences	Click here
Page breaks	Click here
Print permissions	Click here
Page Setup	Click here
Header Setup	Click here
Footer Setup	Click here
Printing Options dialog	Click here
Printing reports	Click here
Printing control drawings	Click here
Printing charts	Click here

Printing features

Printing and paging supports the graphical printout of CM and SCM charts and includes the following areas:

- A Page Setup dialog consisting of the following:
  - Header Setup dialog: A button to invoke a Header Setup dialog containing controls for setting the header background bitmap, header text font, and header text sections.
  - Footer Setup dialog: A button to invoke a Footer Setup dialog containing controls for setting the footer background bitmap, footer text font, and footer text sections.
  - Printing Options dialog: Controls for setting page orientation, paper size, paper source, and page margins. Printing Options also includes chart fit options, page ordering options, and SCM printing options.

- Support for printing the entire project tree or monitoring tree with one command
- Support for printing multiple selected unopened charts in the project tree or monitoring tree

Printing preferences

Printing preferences are stored as GUI preferences in the database. The Control Builder GUI Preferences infrastructure is used to set and get the printing preferences.

Page breaks

Page breaks are of the single selection variety, meaning that only one page break can be selected at a time and selecting one page break de-selects all other page breaks and all other items in the chart. Page breaks are stored as symbol attributes on the CM block for a CM chart and are stored as symbol attributes on each handler block in an SCM, including the main handler block. Both the edit and monitoring versions of SCM/SCM charts can have distinct pagination - each version has its own set of page breaks. Initially, the monitoring version can then be changed by the user, which does not affect the page breaks in the edit version. Page breaks are read in from the database when a chart is opened and are saved when the chart is saved.

Class CPageBreak defines the attributes and methods of a single page break. Class CPageBreakInfo defines the attributes and methods of an array of page breaks. Each document instance (of class CContainerDoc) contains a member variable of class CPageBreakInfo. Pagination information in CPageBreakInfo is stored as an unsorted array of horizontal and vertical page breaks for the entire document (i.e., for all charts in the document, not just for a handler of an SCM, for instance). When a control drawing is displayed, the page breaks are drawn using this unsorted array. However, when a control drawing is printed, the page breaks in this unsorted array are first sorted into two sorted arrays - an array of horizontal page breaks and an array of vertical page breaks for the current chart (e.g., for the current handler being printed in an SCM). From these two sorted arrays, it is then possible to determine, for a particular page, where the origin (upper left-hand corner) of the page is located and also what the x- and y- extents of the page are.

During printing, the printing function calculates what the minimum document size is for the current chart being printed (i.e., what the smallest bounding rectangle is that contains all items in the chart being printed). This information is used by the printing function to prevent printing empty pages in a chart, where the chart is basically empty except for some content in the upper left-hand corner of it.

Method CContainerView::OnPrint is responsible for printing a page of a chart. It prints the header and footer for the page and also the actual contents of the chart page.

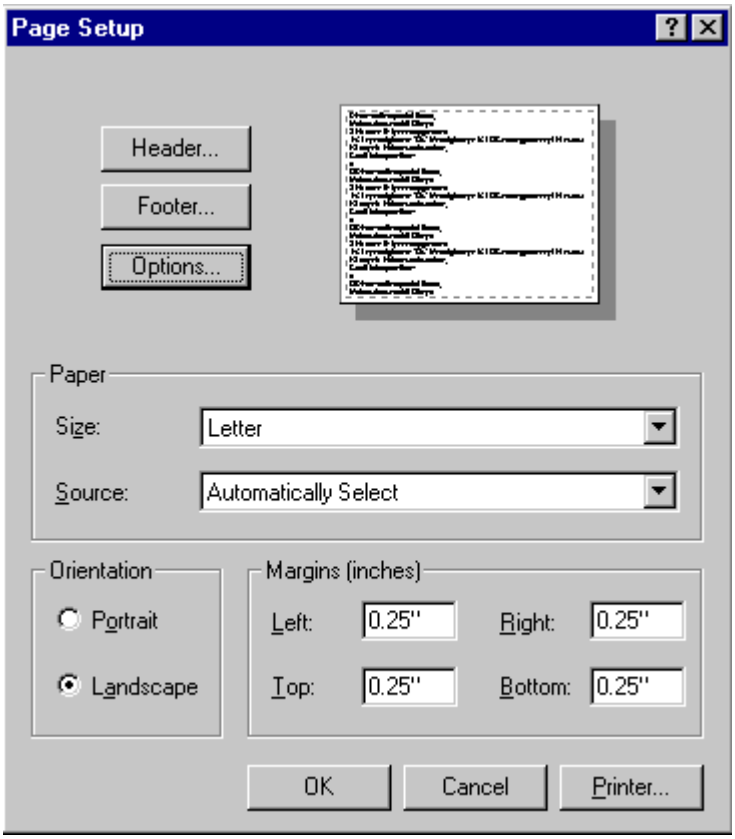
Print permissions

User level	Permissions
LVL1 user and above	May change print settings. Setting changes will not be saved.
ENGR user and above	May change print settings. Setting changes may be saved.

Page Setup

The Page Setup dialog box is used to access the Header, Footer, and Options dialog boxes. It is also used to set up the standard printer features such as paper size and source, as well as margins and orientation.

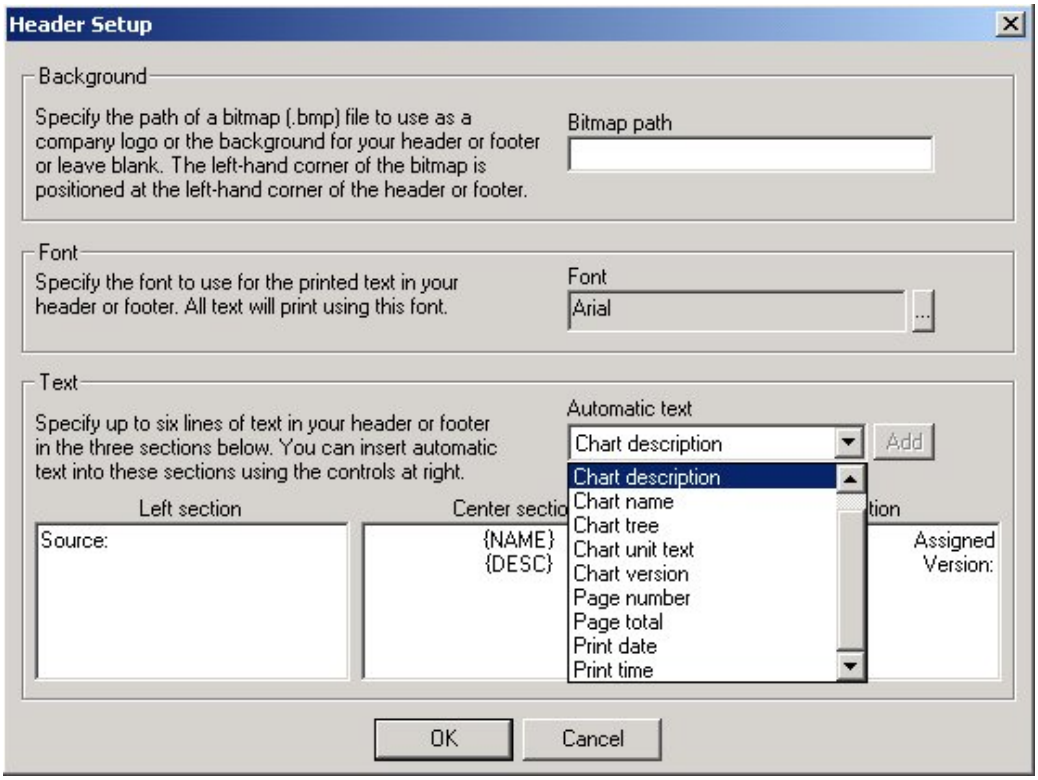
The Page Setup dialog is accessed through File -> Page Setup.



Page Setup dialog Box

Header Setup

The Header Setup dialog is accessed through File -> Page Setup, then selecting the Header... button.



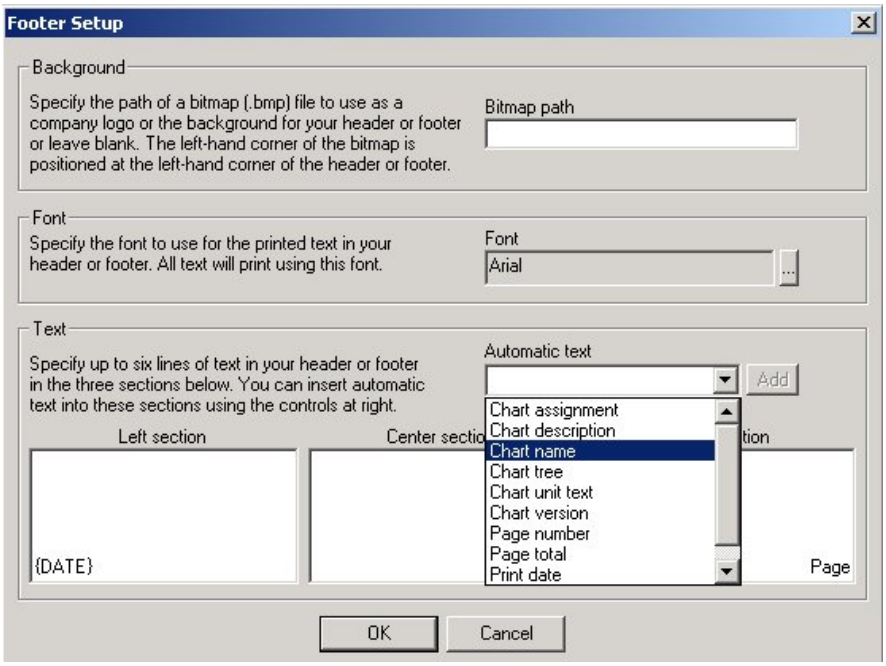
Header Setup dialog box

The Header Setup dialog box allows the user to format the printed documents header appearance.

Setup feature	Use of..
Background	A background graphic may be inserted in the form of a bitmap (.bmp) file by entering the file pathway. The bitmap will always be positioned on the left of the header.
Font	Font styles may be changed, including font family, style, size and underline. Strikeout and style colors are not allowed. The Font dialog box can be accessed by selecting the button next to the font style name.
Text	All automatic text fields can be inserted into the Left, Center, and Right sections and are expanded at runtime. Up to six lines of text may be entered into each of the sections. Text may be entered directly into the section fields, or the text fields can be inserted using the Automatic text drop-down box. To enter a text field, the cursor must be placed into the section field before the drop-down selection is chosen. If more than six lines of text are entered, they are not saved or printed. Lines of text exceeding 255 characters are also not saved.

Footer Setup

The Footer Setup dialog is accessed through File -> Page Setup, then selecting the Footer... button.



Footer Setup dialog box

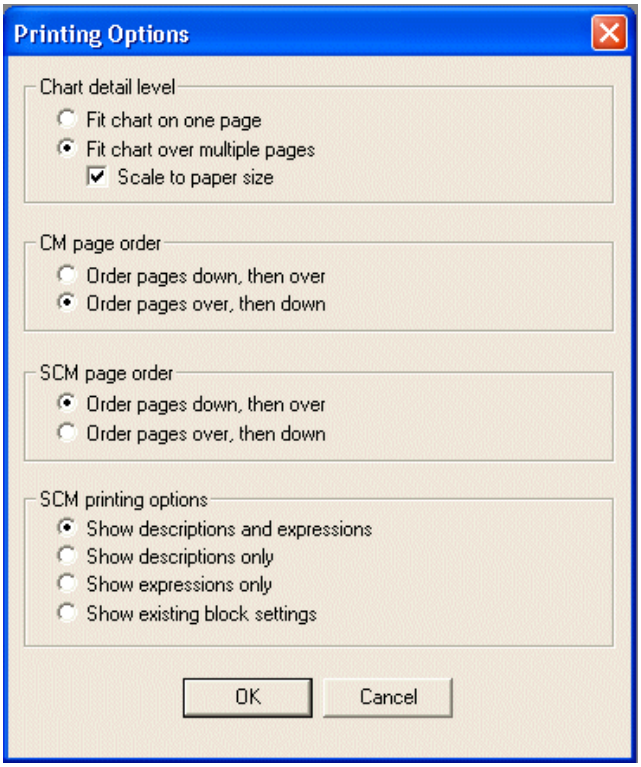
The Footer Setup dialog box allows the user to format the printed documents header appearance. The Footer Setup functions and instructions are nearly identical to the Header Setup.

Setup feature	Use of..
Background	A background graphic may be inserted in the form of a bitmap (.bmp) file by entering the file pathway. The bitmap will always be positioned on the left of the footer.
Font	Font styles may be changed, including font family, style, size and underline. Strikeout and style colors are not allowed. The Font dialog box can be accessed by selecting the button next to the font style name.
Text	All automatic text fields can be inserted into the Left, Center, and Right sections and are expanded at runtime. Up to six lines of text may be entered into each of the sections. Text may be entered directly into the section fields, or the text

Setup feature	Use of..
	fields can be inserted using the Automatic text drop-down box. To enter a text field, the cursor must be placed into the section field before the drop-down selection is chosen. If more than six lines of text are entered, they are not saved or printed. Lines of text exceeding 255 characters are also not saved.

Printing Options dialog

The Printing Options dialog is accessed through File -> Page Setup, then selecting the Options... button.



Printing Options dialog box

The Printing Options dialog box allows the following specifications:

Option	Effect
Chart detail level	Chart may be sized to fit on one, or multiple pages. • A chart having no page breaks will print on one page when the chart detail level is set to "Fit chart on one page". • A chart having page breaks will print on one page when the chart detail level is set to "Fit chart on one page". The chart will be sized to fit on the page. • A chart having no page breaks will print over multiple pages when the chart detail level is set to "Fit chart over multiple pages". In this case, default pagination will occur, similar to how pagination occurred before CB Printing Enhancements was introduced. A chart having page breaks will print over multiple pages when the chart detail level is set to "Fit chart over multiple pages".
Scale to paper size	Check this option to scale the chart to the printer paper size and print the chart within its printable area or will use the default paper size.
CM page order	CM page order - Pages may be ordered by prioritizing over or down first. • Pages in a CM chart will be ordered and printed down then over when the CM page order option is set to "Order pages down, then over", respectively. Refer to Printing Charts for detailed information. • Pages in a CM chart will be ordered and printed over then down when the CM page order option is set to "Order pages over, then down", respectively. Refer to Printing Charts for detailed information.

Option	Effect
SCM page order	Pages may be ordered by prioritizing over or down first. Refer to Printing Charts for detailed information. • Pages in an SCM chart will be ordered and printed down then over when the SCM page order option is set to "Order pages down, then over", respectively. • Pages in an SCM chart will be ordered and printed over then down when the SCM page order option is set to "Order pages over, then down", respectively.
SCM printing options	Select between showing descriptions and/or expressions or existing block settings. • All steps and transitions in an SCM chart will print with their descriptions and expressions (even when the user has hidden the details box) when the SCM printing option is set to "Show descriptions and expressions". • All steps and transitions in an SCM chart will print with their descriptions only (even when the user has hidden the details box) when the SCM printing option is set to "Show descriptions only". • All steps and transitions in an SCM chart will print with their expressions only (even when the user has hidden the details box) when the SCM printing option is set to "Show expressions only". • All steps and transitions in an SCM chart will print exactly as they are displayed when the SCM printing option is set to "Show existing block settings". Hidden details boxes will not print.

Printing reports

Prerequisites:

- Control Builder is running
- Tree windows are open

Step	Action	Result
1	Select File -> Print -> Reports.	Report dialog appears on the screen.
2	Select the report type, and choose from the available IOMs, CMs and SCMs.	
3	Click the appropriate button for the desired operation (such as print, preview, export).	The desired operation executes.
4	Click the Close button to close the Reports dialog	

Printing control drawings

File -> Print -> Control Drawings brings up the basic printer dialog box. Any formatting or selection of print options must be done through File -> Page Setup prior to printing.

Printing charts

Multiple selected up-opened charts can be printed from either the project tree or the monitoring tree using File -> Print -> Charts. Any formatting or selection of print options must be done through File -> Page Setup prior to printing.

Type of chart to print	Command
All project tree charts (whether the project tree is open or not)	File -> Print -> All Project Tree Charts.
All monitoring tree charts (whether the monitoring tree is open or not)	File -> Print -> All Monitoring Tree Charts

The CM control drawing pages are defined as follows.

CM page order when option is "Order pages over, then down"

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

CM page order when option is "Order pages down, then over"

1	6	11	16	21
2	7	12	17	22
3	8	13	18	23
4	9	14	19	24
5	10	15	20	25

The SCM control drawing pages are set up in a similar manner, except that there are only 3 columns of pages and 10 rows by default.

Step	Action	Result
1	With the desired Control Drawing chart open on the screen, select File -> Print -> Chart. Alternate method: Select <Ctrl><P>.	Print dialog appears on the screen.
2	Make the desired selections (such as, name, print range, number of copies) on the Print dialog and click OK.	

An entire chart must be printed. It is not possible to print only specific areas of a chart.

Using the Regulatory Loop Wizard

The Regulatory Loop Wizard allows you to quickly build a standard default Regulatory Loop Control Module using Control Builder. The Wizard includes a Suggest Tuning Constants option which provides estimated initial controller tuning values based on basic knowledge of the loop process requirements. This option may be bypassed, if preferred, so that known tuning values may be entered.

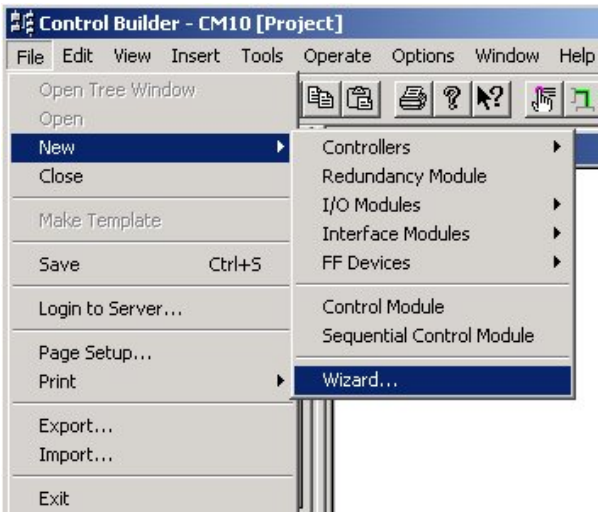
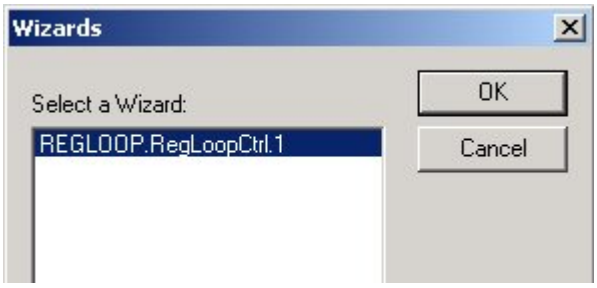
Prerequisites:

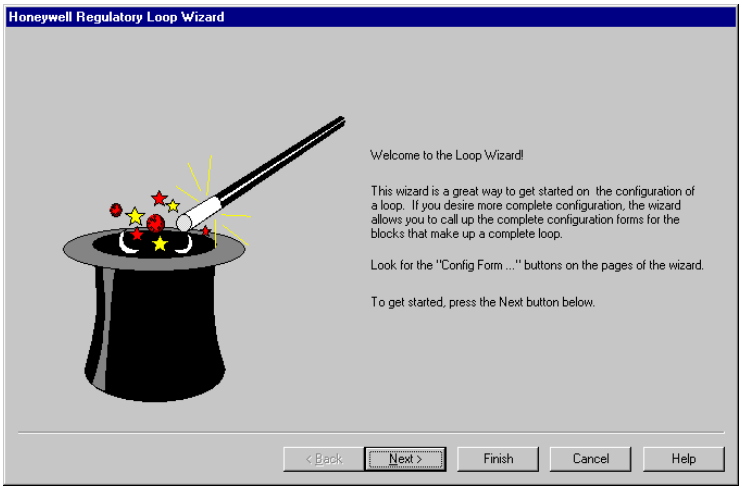
- Control Builder is running
- Tree windows are open

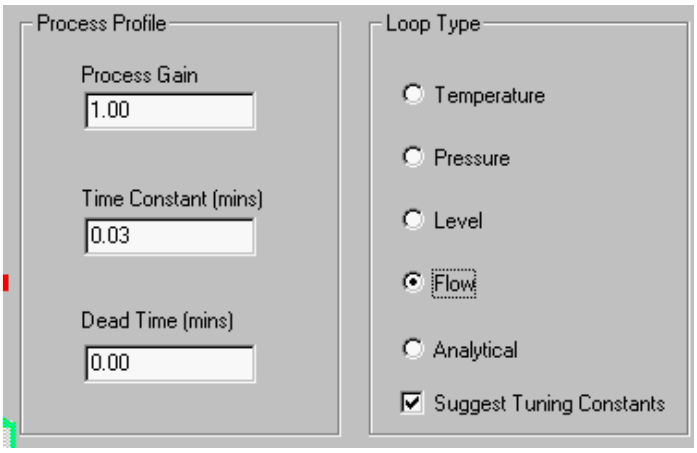
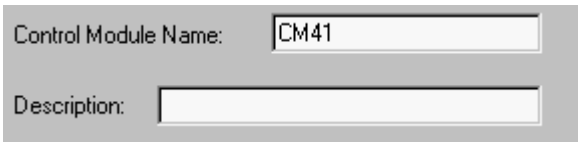
Considerations:

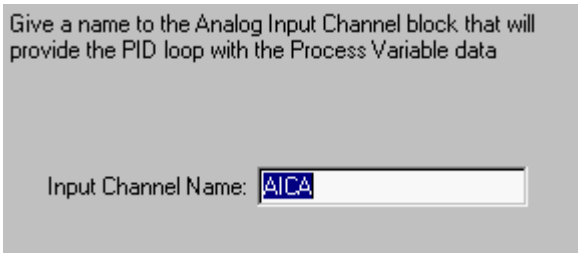
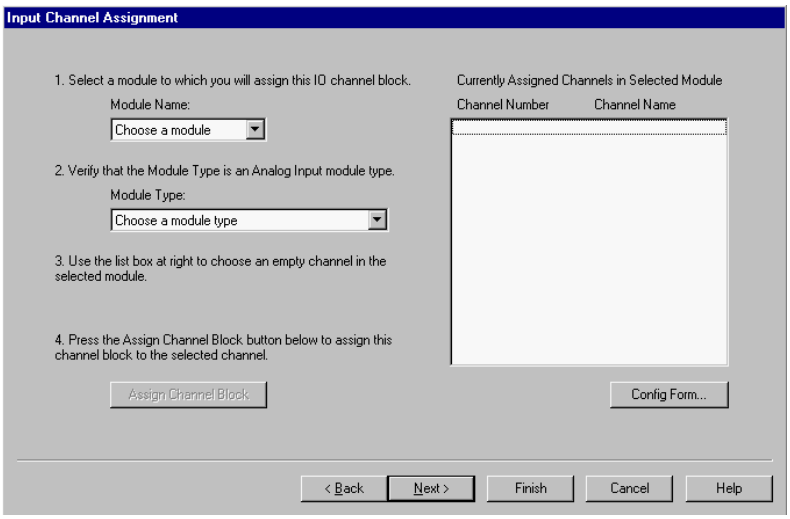
- All illustrations used in the procedure are for example purposes only.

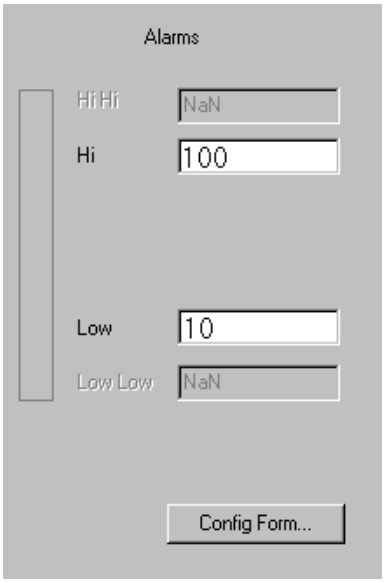
Perform the following steps to use the Regulatory Loop Wizard:

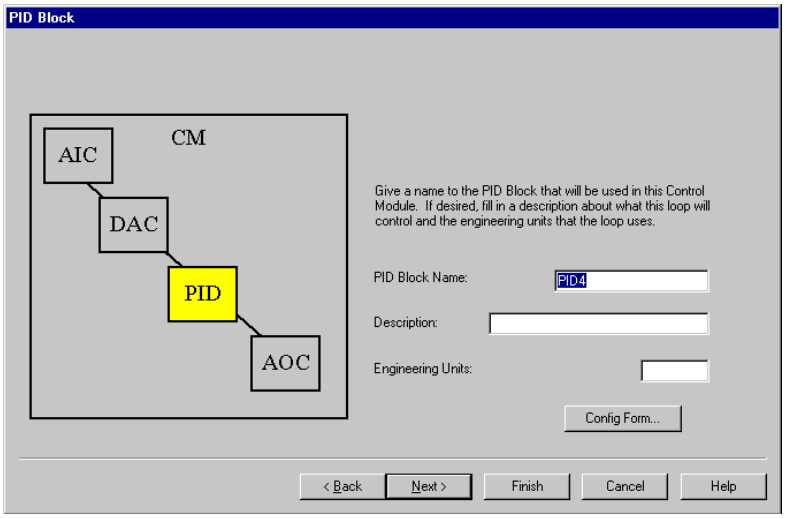
Step	Action
1	Click File -> New ->Wizard... . 
2	Select REGLOOP.RegLoopCtrl.1 from the Wizards selection dialog that appears on the screen and click OK. 

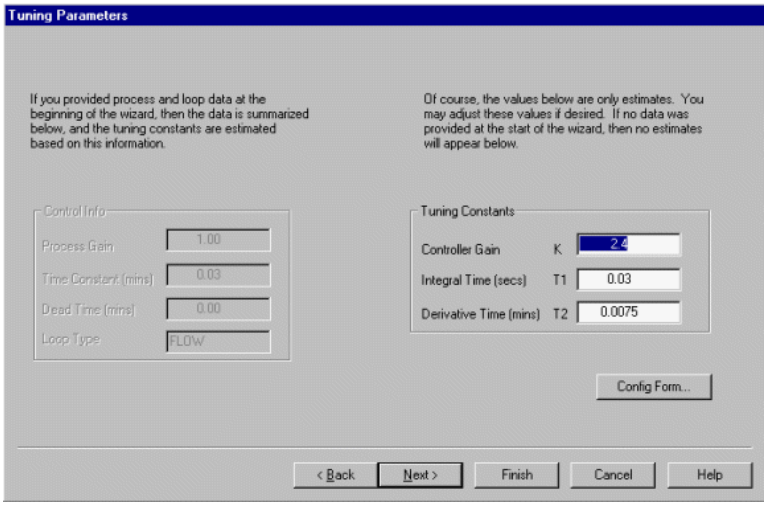
Step	Action
3	The Regulatory Loop Wizard Welcome page appears on the screen. Read the introductory information and then click Next. 

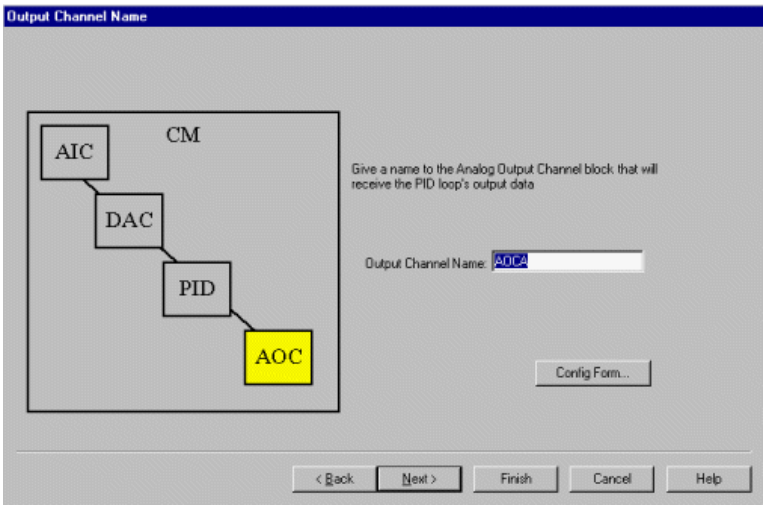
Step	Action
4	The Process Profile page appears on the screen. Enter the indicated process information along with the type of loop needed to control the process, then click Next.  ***Note:*** The Suggest Tuning Constants, when selected, provides ballpark estimates for initial controller tuning values based on the data parameters you have already entered in the Process Profile.
5	The Control Module Name page appears on the screen. Enter a name and description for the Control Module that will be used to contain the loop's function blocks and control configuration, then click Next. Note: Select the Config. Form button in this and any subsequent pages if desired to access the Parameters Configuration form for the respective function block and thereby enter more detailed parameter configuration preferences. 

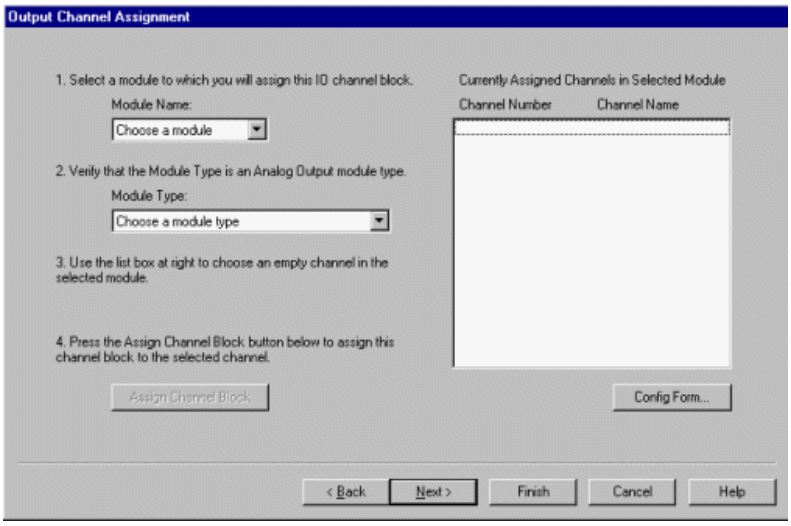
Step	Action
6	The Input Channel Name page appears on the screen. Enter a name for the Analog Input Channel block that will provide the regulatory loop with the process variable data, then click Next. 
7	The Input Channel Assignment page appears on the screen. If necessary, enter the requested information to assign the appropriate Analog Input (AI) Module to the AICHANNEL block that is associated with this regulatory loop. Then click Next. 
8	The Input Channel Configuration page appears on the screen. If you wish to configure on the Analog Input Module the input channel that was just created, press the Module Configuration button and enter the requested configuration parameters. Then return to the Input Channel Configuration page and click 

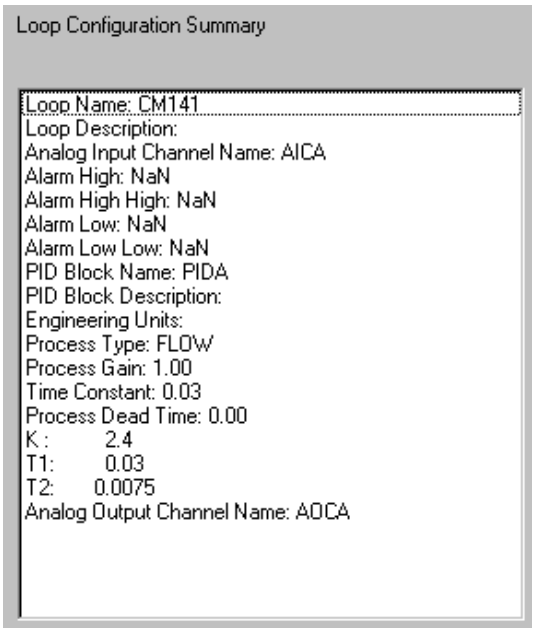
Step	Action
9	The Alarming page appears on the screen. Enter desired alarm limit values for the available high and low alarms and then click Next. 

Step	Action
10	The PID Block page appears on the screen. Enter the requested configuration parameters for the PID block and then click Next. 

Step	Action
11	The Tuning Parameters page appears on the screen. Based on what was entered previously, the process and loop data is summarized on this page and estimated tuning constants are presented. Adjust the tuning constants as desired, then click Next. 

Step	Action
12	The Output Channel Name page appears on the screen. Enter a name for the Analog Output Channel (AOCHANNEL) block that is to receive output data from the PID block in the Regulatory Loop. Then click Next. 

Step	Action
13	The Output Channel Assignment page appears on the screen. If necessary, enter the requested information to assign the appropriate Analog Output (AO) Module to the AOCHANNEL block that is associated with this Regulatory Loop. Then click Next. 
14	The Output Channel Configuration page appears on the screen. If you wish to configure on the Analog Output Module the output channel that was just created, press the Module Configuration button and enter the requested configuration parameters. Then return to the Output Channel Configuration page and click Next >.

Step	Action
15	The Loop Wizard Summary page appears on the screen. The data that has been entered in the Regulatory Loop Wizard is summarized in the Loop Configuration Summary box. Review the data to ensure that the desired configuration parameters have been entered. Any desired changes can be made by backing up to the page in the Wizard where the erroneous information was entered and making adjustments there. Then return to this page and click Next. 
16	The Finished! Page appears on the screen. Click on the Finish button at the bottom of the page and continue with the complete configuration of your control loop.

Exporting function block configurations

This section includes information that you can reference for:

Topic	Link
Some general export considerations	Click here
Opening the Export dialog box	Click here
Exporting a selected project	Click here
Exporting selected function block(s)	Click here
Exporting a particular type of function block	Click here
Canceling an export currently in progress	Click here
Commencing an Export while a load is in progress	Click here
Commencing a Load while an Export is in progress	Click here
Exporting data to a user selected directory	Click here

Some general export considerations

The following are considerations when exporting function block configurations.

REFERENCE - INTERNAL

For more information on Exporting Function Block Configurations refer to Experion Control Builder Components Theory, Control Builder Export and Import Functionality.

ATTENTION

- Import and Export operate only on the primary database. Operation on the secondary database is not permitted.
- User selection of Import as well as Export should not be permitted if Engineering Repository synchronization is in progress.

CAUTION

During the export function, a file is created by the system called Export.sl. This file contains the list of objects to be exported. If the file exists in the specified directory, a subsequent export function will overwrite it.

Opening the Export dialog box

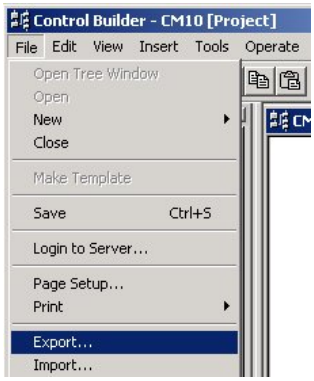
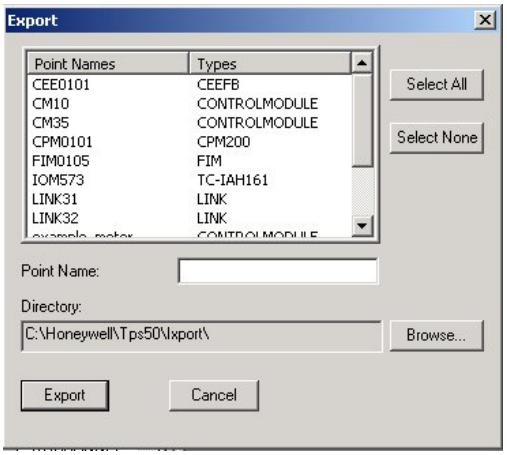
Prerequisites:

- Control Builder is running
- Tree windows are open

Considerations:

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

Follow the step in the table below to open the export dialog box.

Step	Action	Result
1	From Control Builder, click File -> Export to open the Export dialog box. 	The Export dialog box appears. 

Exporting a selected project

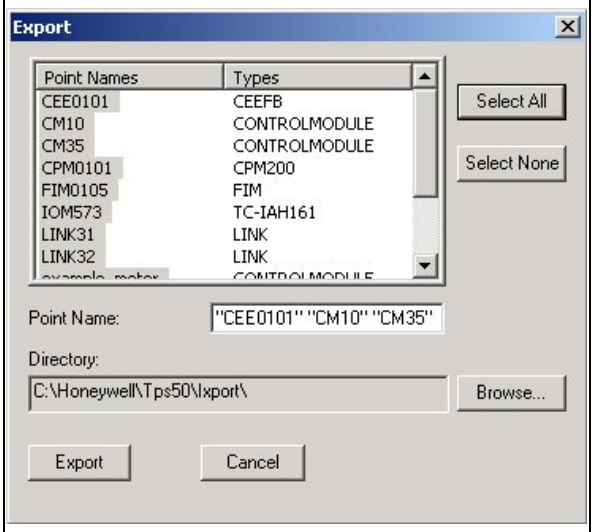
Prerequisites:

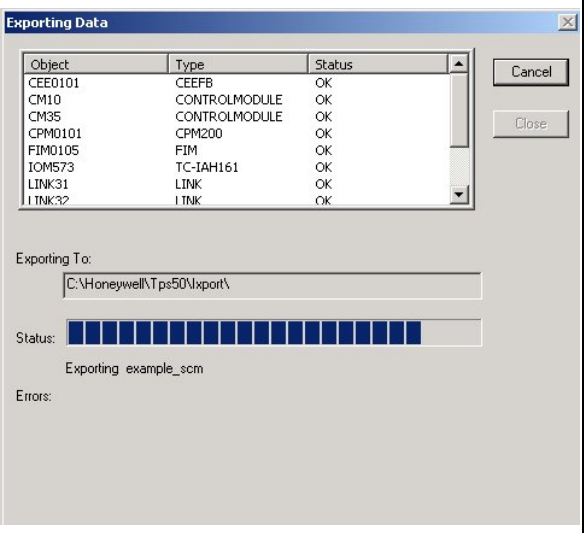
- Control Builder is running
- Tree windows are open

Considerations:

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

Follow the steps in the table below to export a selected project.

Step	Action	Result
1	From the Export dialog box, click **Select All** to select all of the function blocks.	The selected function blocks are highlighted on the dialog box. 

Step	Action	Result
2	Click **Export** to export the selected function blocks.	The Exporting Data dialog box appears.  When the Exporting Data dialog box disappears, the export is done.

Exporting selected function block(s)

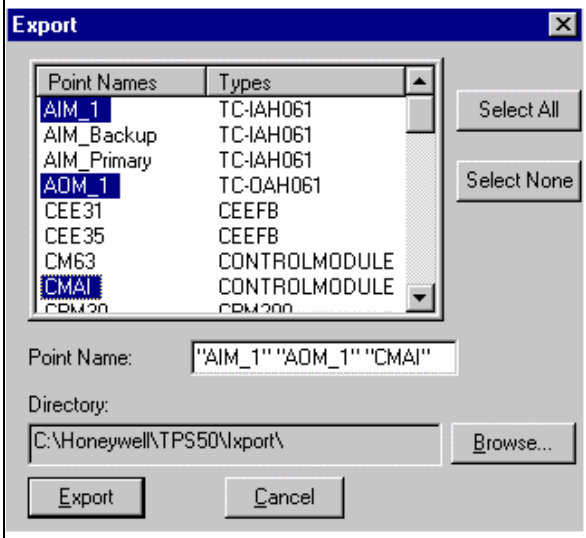
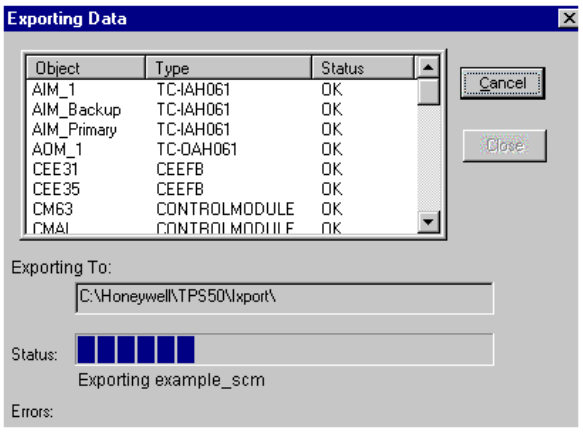
Prerequisites:

- Control Builder is running
- Tree windows are open

Considerations:

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

Follow the steps in the table below to export selected object(s).

Step	Action	Result
1	From the Export dialog box, press <Control> and click the function block(s) you want to export.	The selected function block(s) are highlighted on the dialog box. 
2	Click **Export** to export the selected function blocks.	The Exporting Data dialog box appears.  When the Exporting Data dialog box disappears, the export is done.

Exporting a particular type of function blocks

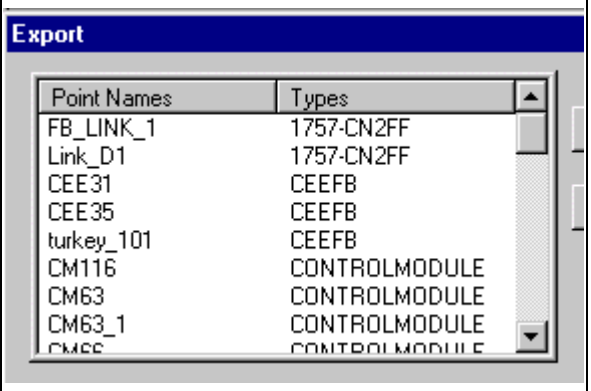
Prerequisites:

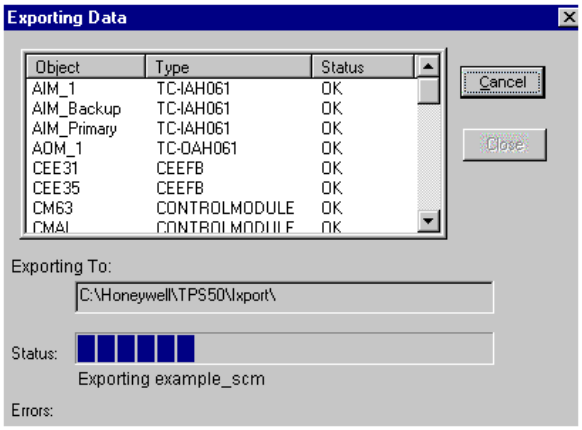
- Control Builder is running
- Tree windows are open

Considerations:

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

Follow the steps in the table below to export a particular type of object.

Step	Action	Result
1	From the Export dialog box, click the column header, **Types** to sort the function blocks by types	The function blocks will be sorted by type. 

Step	Action	Result
2	Click **Export** to export the selected function blocks.	The Exporting Data dialog box appears.  When the Exporting Data dialog box disappears, the export is done.

Canceling an export currently in progress

Prerequisites:

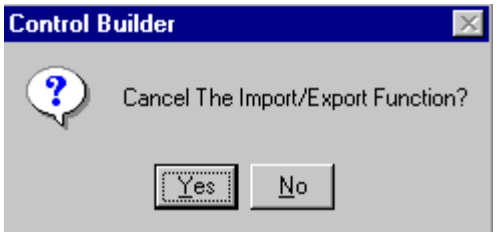
- Control Builder is running
- Tree windows are open

Considerations:

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

Follow the steps in the table below to cancel an export currently in progress.

Step	Action	Result
1	From the Exporting Data dialog box, click **Cancel.**	The following dialog box will appear.

Step	Action	Result
		
2	Click **Yes** to cancel the export.	The export is canceled.

Commencing an Export while a Load is in progress

REFERENCE - INTERNAL

For more information on loading a control strategy, refer to Control Strategy Loading.

ATTENTION

- An Export may not be commenced during a load which has been triggered from the Controller Assignment dialog box.
- Load has a "stay on top" feature, so you may need to move windows around to interact with the Export dialog box.

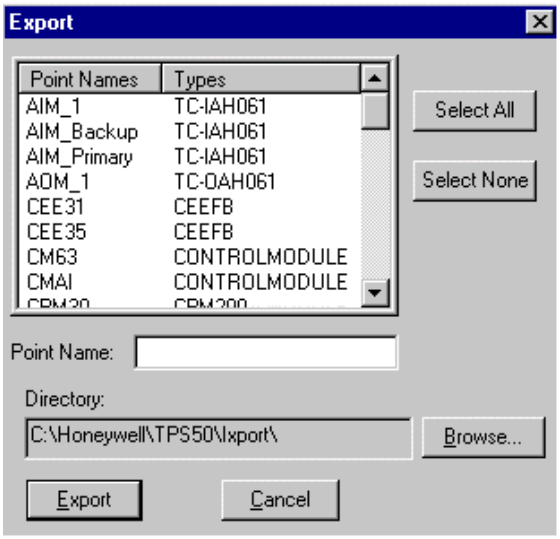
Prerequisites:

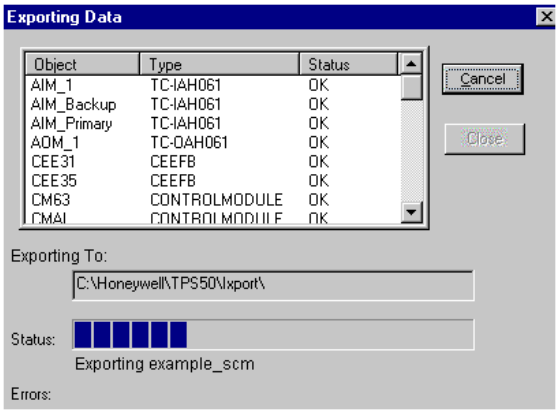
- Control Builder is running
- Tree windows are open

Considerations:

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

Follow the steps in the table below to commence an Export while a LOAD is in progress.

Step	Action	Result
1	With a Load in progress, from Control Builder, click **Export** to open the Export dialog box.	The Export dialog box appears. 
2	Select the object(s) you wish to export.	
3	If desired, browse to the directory you want to export to.	

Step	Action	Result
4	Click **Export.**	The Exporting Data dialog box appears.  When the Exporting Data dialog box disappears, the export is done.

Commencing a Load while an Export is in progress

ATTENTION

Load has a "stay on top" feature, so you may need to move windows around to interact with the Export dialog box.

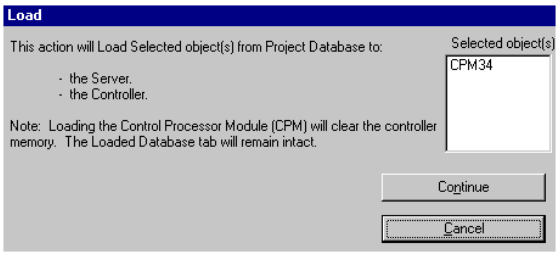
Prerequisites:

- Control Builder is running
- Tree windows are open

Considerations:

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

Follow the steps in the table below to commence a Load while an Export is progress.

Step	Action	Result
1	With an export in progress, click the  icon to commence a load. Alternatively, commence the load from the Assignment dialog.	The Load dialog box will appear. 
2	Click **Continue** to commence the load.	When the Load dialog box disappears, the load is complete.

Exporting data to a user selected directory

Prerequisites:

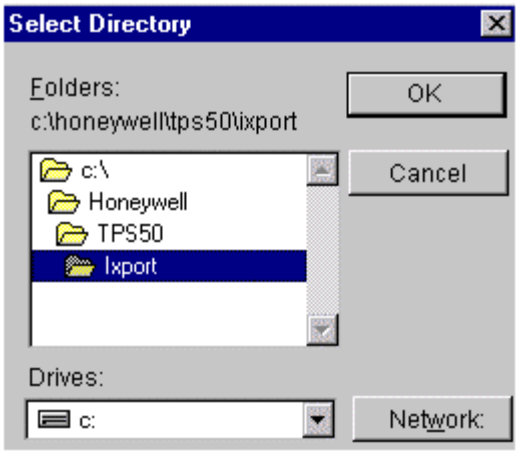
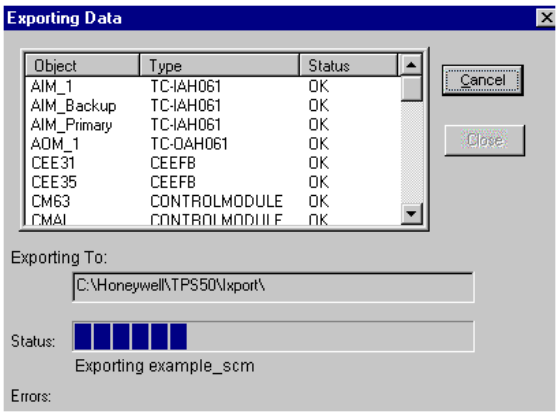
- Control Builder is running
- Tree windows are open

Considerations:

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

Follow the steps in the table below to export data a user selected directory.

Step	Action	Result
1	From the Export dialog box, click **Browse**.	The Select Directory dialog box appears:

Step	Action	Result
		
2	Browse to the desired directory.	
3	Click **OK**.	
4	Click **Export.**	The Exporting Data dialog box appears.  When the Exporting Data dialog box disappears, the export is done.

Importing function block configurations

This section includes information that you can reference for:

Topic	Link
Some general import considerations	Click here
Opening the Import dialog box	Click here
Importing function blocks from selection list	Click here
Importing function blocks from directory	Click here
Canceling an import currently in progress	Click here
Commencing an Import while a Load is in progress	Click here
Commencing a Load while an Import is in progress	Click here

Some general import considerations

CAUTION

- During the import function, a file is created by the system called Import.sl. This file contains the list of objects to be imported. If the file exists in the specified directory, a subsequent import function will overwrite it.
- During the import function, imported blocks write over each block, but new blocks are left as is.

The following table summarizes some things you should consider before initiating the import function. This list is by no means comprehensive and should be supplemented with any first hand experience gained from previous imports. As a rule of thumb, it is better to import a large database in small logical segments, such as importing all IOMs first, rather than all at once.

If Database to be Imported Includes . . .	Then, Consider Doing This . . .
Example Control Modules and Sequential Control Modules provided with the "Clean" database	Delete the Example Control Modules and Sequential Control Modules from the "Clean" database that is to receive the import or do not import the example modules.
Fieldbus devices	Build templates for the devices using the File->New->Type->Fieldbus Device menu selection and import the fieldbus devices first before importing any Control Modules (CMs). Otherwise, a fieldbus device import will fail if its corresponding device template is not present. Likewise, import of a CM containing a fieldbus function block will fail if its corresponding device has not yet been imported
Input/Output Modules (IOMs) and/or Input/Output Processors (IOPs)	Import IOMs and IOPs first before importing any Control Modules. Otherwise, import of a CM containing an IOM/IOP channel function block will fail if its corresponding IOM/IOP block has not yet been imported.

Opening the Import dialog box

Prerequisites:

- Control Builder is running
- Tree windows are open

Considerations:

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

Follow the steps in the table below to open the Import dialog box.

Step	Action	Result
1	From Control Builder, click File -> Import to open the Import dialog box.	The Import dialog box appears.

Importing function blocks from selection list

Prerequisites:

- Control Builder is running
- Tree windows are open

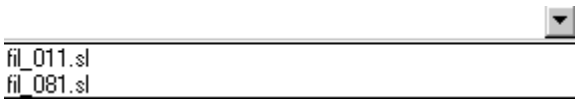
Considerations:

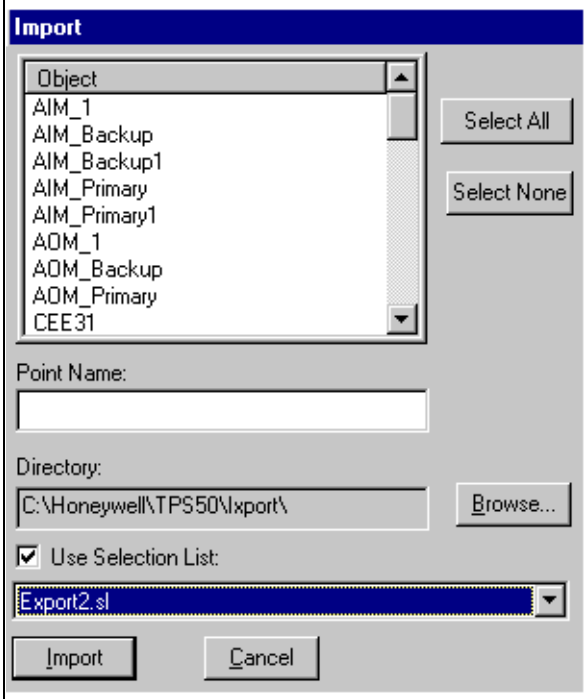
- All illustrations used in the procedure are for example purposes only.

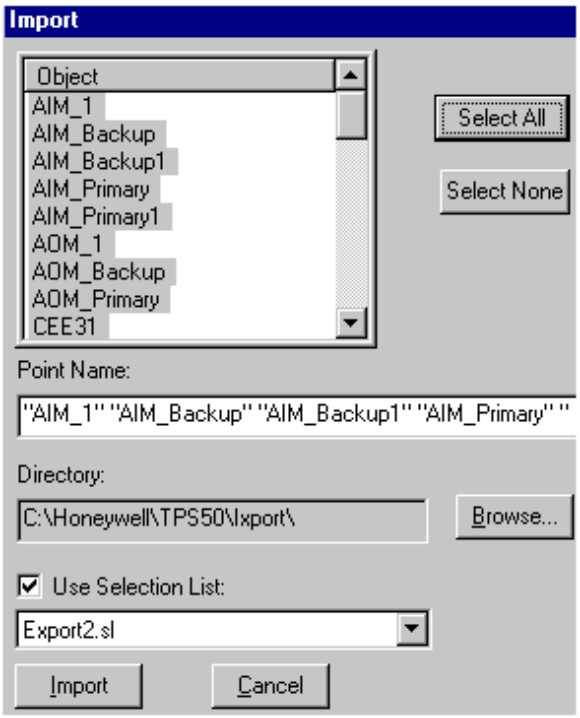
ATTENTION

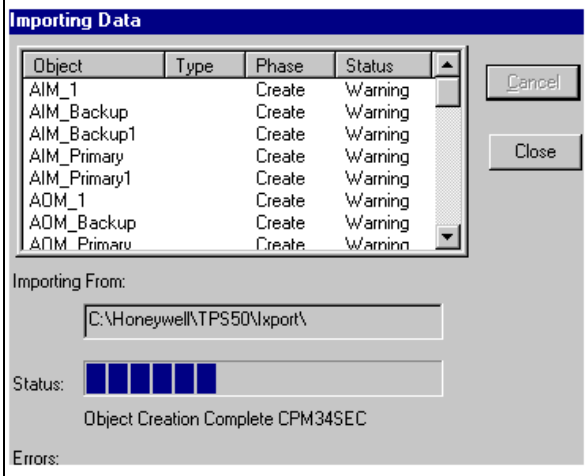
Using the Notepad Accessory, you may edit the existing selection list, Export.sl, to create a new selection list of function blocks you wish to import.

Follow the steps in the table below to import function blocks from a selection list.

Step	Action	Result
1	From the Import dialog box, click Use Selection List check box. 	
2	Click  to view the names of selections lists in the directory.	The names of the selection lists will appear. 

Step	Action	Result
3	Select the selection list file from the combo box.	A list of the function blocks in the selection list appears in the dialog box. 

Step	Action	Result
4	Select the function blocks you wish to import. If you wish to import all of the function blocks in the selection list, click **Select All**. Note: You must have object(s) selected in order to do the import.	The following figure illustrates the result of clicking **Select All**. 

Step	Action	Result
5	Click **Import**.	The Importing Data dialog box appears.  When the Importing Data dialog box disappears, the import is done.

Importing function blocks from directory

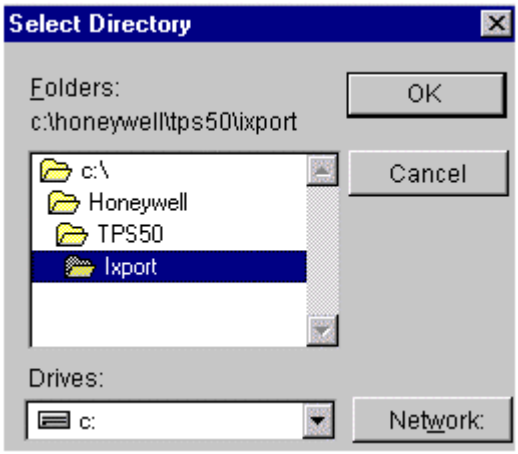
Prerequisites:

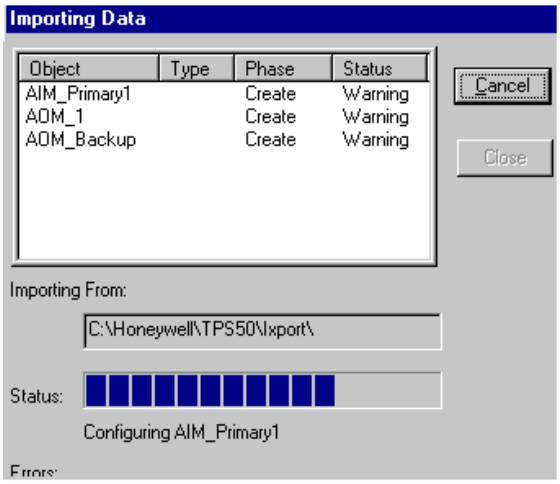
- Control Builder is running
- Tree windows are open

Considerations:

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

Follow the steps in the table below to import function blocks from a directory.

Step	Action	Result
1	From the Import dialog box, click **Browse**.	The Select Directory dialog box appears: 
2	Browse to the desired directory.	
3	Click **OK** to commence the import.	
4	Click **Import** to import the function blocks.	The Importing Data dialog box appears.

Step	Action	Result
		 When the Importing Data dialog box disappears, the import is done.

Canceling an Import currently in progress

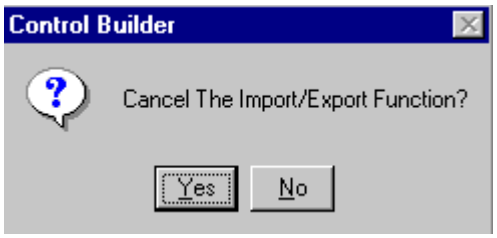
Prerequisites:

- Control Builder is running
- Tree windows are open

Considerations:

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

Follow the steps in the table below to cancel an Import currently in progress.

Step	Action	Result
1	From the Import dialog box, click **Cancel**.	The following dialog box will appear. 
2	Click **Yes** to cancel the import.	The import is canceled.

Commencing an Import while a Load is in progress

ATTENTION

- An import may not be commenced during a Load which has been triggered from the Controller Assignment dialog box.
- Load has a "stay on top" feature, so you may need to move windows around to interact with the Import dialog box.

Prerequisites:

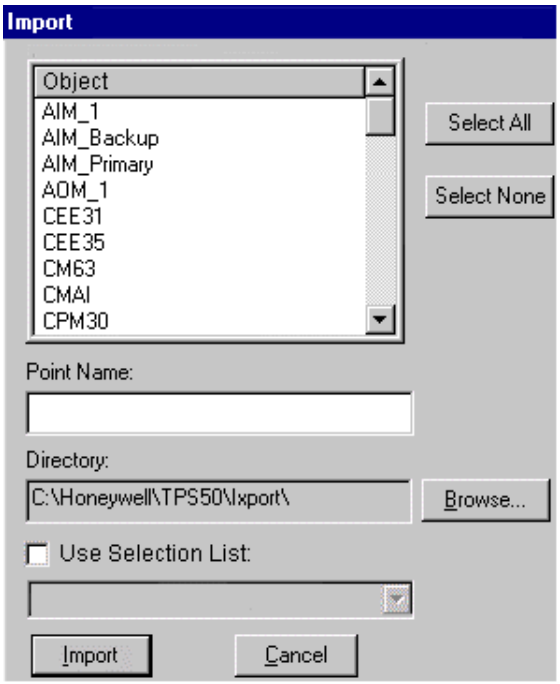
- Control Builder is running
- Tree windows are open

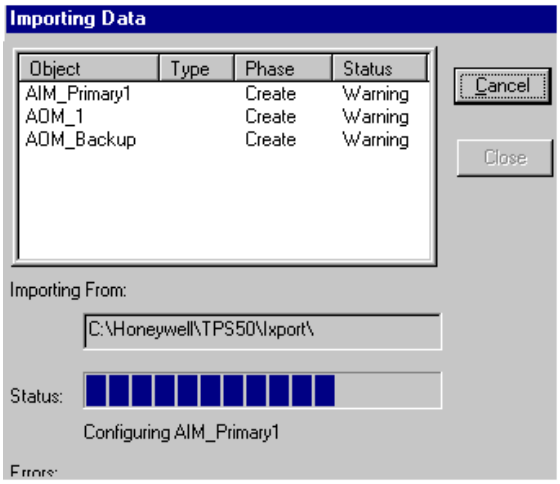
Considerations:

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

Follow the steps in the table below to cancel an import while a Load is in progress.

Step	Action	Result
1	With a Load in progress, from Control Builder, click **Import** to open the Export dialog box.	The Import dialog box appears.

Step	Action	Result
		
2	Select the function block(s) you wish to import.	
3	If desired, browse to the directory you want to export to.	
4	Click **Import.**	The Importing Data dialog box appears.

Step	Action	Result
		 When the Importing Data dialog box disappears, the import is done.

Commencing a LOAD while an Import is in progress

ATTENTION

Load has a "stay on top" feature, so you may need to move windows around to interact with the Export dialog box.

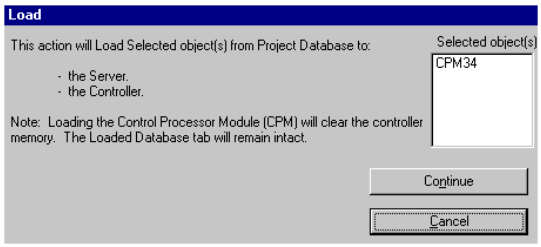
Prerequisites:

- Control Builder is running
- Tree windows are open

Considerations:

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

Follow the steps in the table below to commence a LOAD while an Import is in progress.

Step	Action	Result
1	With an import in progress, click the icon to commence a load. Alternatively, commence the load from the Assignment dialog.	The Load dialog box will appear. 
2	Click **Continue** to commence the load.	When the Load dialog box disappears, the load is complete.

Using Module Hierarchy

This section includes information that you can reference for:

Topic	Link
Purpose	Click here
Initializing Module Hierarchy	Click here
Printing the Module Hierarchy Chart	Click here
Module Hierarchy reports	Click here

Purpose

The purpose of Module Hierarchy is to provide a more hierarchical view of the Control Builder configuration and it's contained children.

For information relating to Detail Displays, refer to the *HMIWeb Display Building Guide*.

Initializing Module Hierarchy

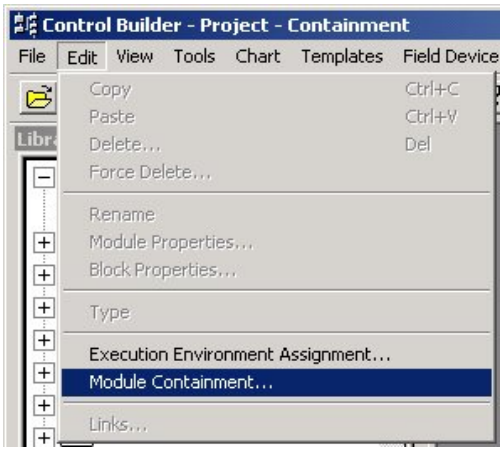
Prerequisites:

- Control Builder is running
- Tree windows are open

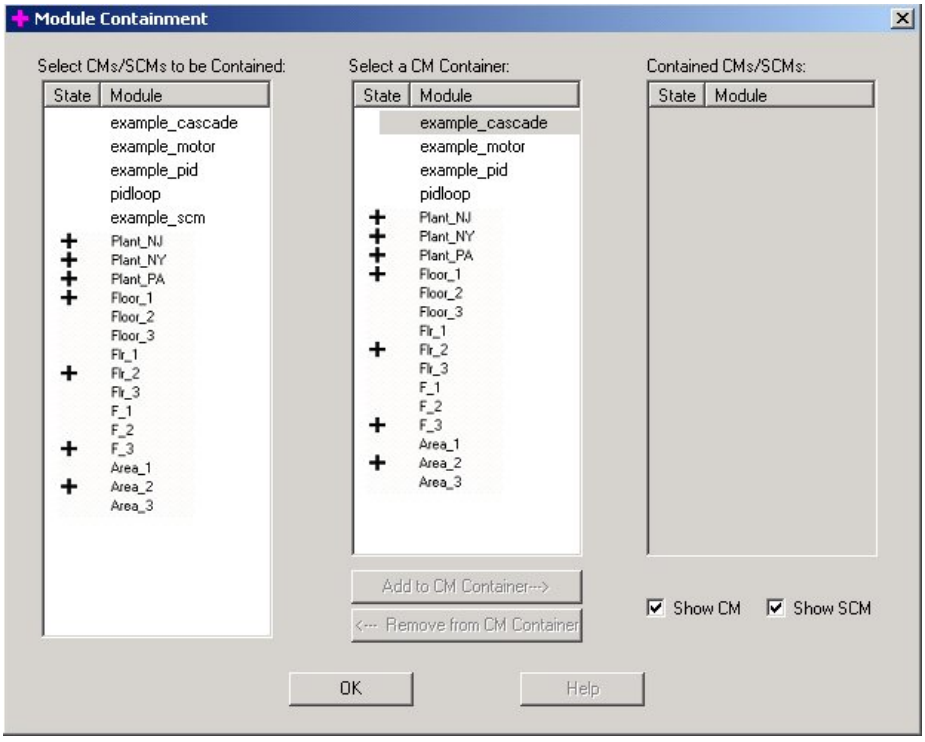
Considerations:

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

Step	Action
1	Start Control Builder.
2	Select Edit -> Module Containment… to open the Module Containment dialog box.



Initializing Module Containment



Module Containment dialog box

CMs with Containment children will have a "+" marker in the State column. CMs without Containment children and all SCMs will not have the marker.

The right side list shows the Containment children (CMs and/or SCMs) of the selected CM. These may be removed from the CM by selecting the Remove from CM button.

Unchecking the 'Show CM' and 'Show SCM' toggles hides the CMs or SCMs from the lists.

Clicking 'Module' at the top of each list will resort the module list alphabetically, toggling between forward and reverse order. Clicking 'State' will sort the list by Children status



TIP

The lists of CMs and Modules are loaded when the dialog box is opened. If new CMs/SCMs are added from another Control Builder session while the dialog box is open, these items do not appear in "real time" in the dialog box.

Printing the Module Hierarchy chart

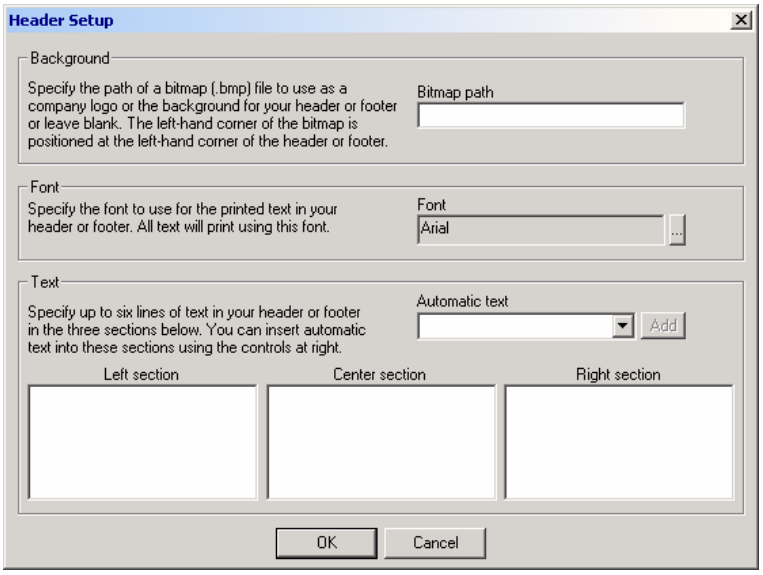
Prerequisites:

- Control Builder is running
- Tree windows are open

Considerations:

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

Step	Action
1	Access the Reports dialog box through File -> Page Setup…
2	Click either the Header or Footer buttons, depending on your preferences.

Step	Action
	
3	"Parent container" will appear in the "Automatic text" drop down list box. Select the item and click into one of the section edit boxes.
4	Click the "Add" button and {PARENT} is pasted into the selected edit box.

TIP

The identical steps are used in the Footer Setup dialog box.

Module Hierarchy reports

The report will contain the following information:

- A title on each page identifying it as a “Containment Report”.
- For each “Contained Child” the following data attributes will appear:
  - Description
  - Controller Assignment
  - Area
  - Parent. (Parent appears twice on the Report Page.)
- Table headings: Name, Description, Controller Assignment, Area, and Parent.
- The time and date when the report was created.
- Page numbers for multiple pages

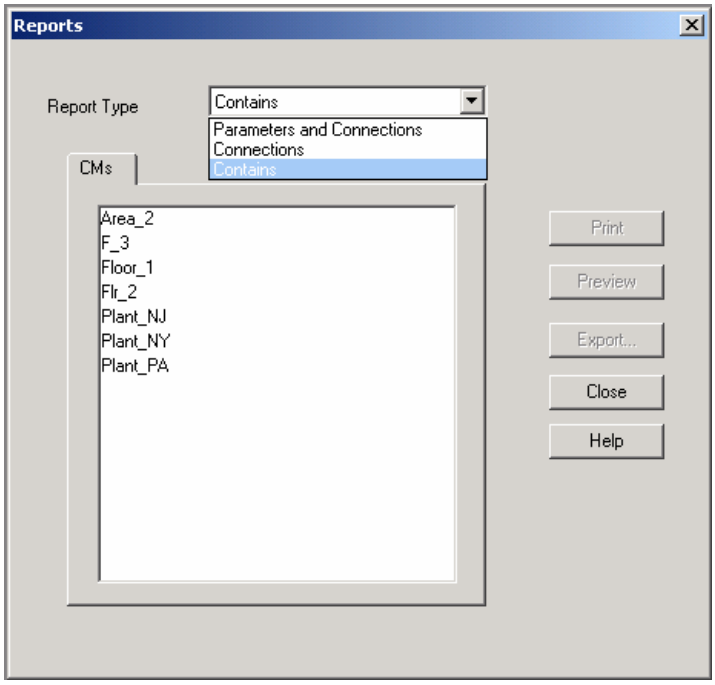
Prerequisites:

- Control Builder is running
- Tree windows are open

Considerations:

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

Accessing the Module Hierarchy report

Step	Action
1	Access the Reports dialog box through File -> Print -> Reports…
	
2	In the Report Type drop-down box, select Contains.
3	Select a CM from the CMs window that you wish to create a report for.
4	Choose to Print, Preview, or Export the report using the buttons on the right.

Working with Profit Loop PKS

This section includes information that you can reference for:

Topic	Link
What is Profit Loop PKS?	Click here
Converting a PID-based control loop to PID-PL	Click here
Reverting to PID block	Click here
Profit Loop PKS Assistant overview	Click here
Starting the Assistant	Click here
Working with Profit Loop PKS Assistant displays	Click here
Obtaining basic information	Click here
Defining a model with Profit Loop PKS Assistant	Click here
Define Model by Direct Entry	Click here
Defining a Model from PID Tuning	Click here
Defining a Model by Loop Type	Click here
Defining a Model Step Testing	Click here
Download a model	Click here
Reverting to a previous model	Click here
Exporting and importing model definitions	Click here
Copying model definitions	Click here
Tune a PID-PL-based controller	Click here
Troubleshooting a PID-PL-based controller	Click here
Controlling access to Profit Loop PKS	Click here

What is Profit Loop PKS?

Profit Loop PKS is a robust, model-based, predictive controller and optimizer for single-input, single-output applications. In many cases, Profit Loop PKS can (and should) be used in place of the standard PID algorithm.

Implementing Profit Loop PKS

Within Control Builder, Profit Loop PKS is represented by the PID-PL block. As such, it is implemented using Control Builder, with basically the same procedures as with other control strategies. Basic procedures in *Creating a control module* apply, in terms of adding a PID-PL function block to your control module.

Procedures specific to Profit Loop PKS are presented in this section.



REFERENCE - INTERNAL

For details on the PID-PL function block, refer to Control Builder Components Theory, Regulatory Control section, PID-PL (Profit Loop PKS) Block.

Converting a PID-based control loop to PID-PL

To convert an existing PID-based control module, with working PID constants, to a PID-PL-based control module, complete the following steps.



ATTENTION

Make sure you follow this procedure. For an accurate conversion, you *cannot* simply change the PID function block's equation type.

Conversion phase

Prerequisites:

- Control Builder is running
- Tree windows are open

Considerations:

- If you will be downloading the converted function block to the control module at the end of this procedure, make sure the controller is offline.
- If you will be downloading later, then the controller does not have to be offline for this conversion procedure.
- Be sure you have installed the Profit Loop Assistant software before attempting a conversion, if you want the PID tuning parameters converted.
- If you wish to convert the existing PID tuning parameters to PID-PL models, a prompt will ask you to define if selected PIDs are Inegrator or Non-Integrator, when you initiate the PID conversion. If you select Integrator PIDs, you must also enter the Closed Loop Response Time. The information you enter applies to ALL of the PIDs selected for conversion. In this case, you should prepare groups of PIDs for conversion before initiating the conversion. You can do this by first differentiating between Integrator and Non-Integrator PIDs and then grouping the Integrator PIDs by the Closed Loop Response Time that applies to the PIDs. You can then convert the resulting groups of PIDs together

Complete the following steps to convert one or more PID function blocks to PID-PL blocks.

Step	Action
1	In Control Builder, make sure that the control module (CM) that contains the PID block to be converted is not open in the control chart for either the **Project** or **Monitoring** view.
2	Click in the **Project** tab that contains the control module(s) to be converted.
3	On the **File** menu, click **Convert PID to PID-PL**.

Step	Action
4	Complete the following steps in the **Convert PID to PID-PL** dialog box: 1) In the **Points** list, select one or more points (control modules, or CMs) to be converted. TIP: To select more than one CM, use Shift+click to select a set of adjacent names, or Ctrl+click to select nonadjacent names. 1) If desired, review your selections in the **Point Name** field. If the entries exceed this field's size, click in the field and use the right and left arrow buttons to scroll in the field. 1) It is a good idea to select a different location for the conversion files, click **Browse**, navigate to the desired location, then click **OK**. For information on the files involved, see *Conversion files* at the end of this procedure. 1) If the PID function block has tuning parameters that you would like to use for a starting model for your PID-PL block, select **Convert PID Tuning Parameters** check box. This selection activates the **Integrator Models** options. If the underlying processes for the selected PIDs are integrating (not self-correcting), select the **Integrator Model** check box and enter the desired **Closed Loop Response Time**. 1) If you want a name other than the original block name, clear the **Retain PIDA Block name** check box. For details about names, see *Naming considerations* at the end of this procedure. 1) Click the **Convert** button.
5	When the conversion is completed, the **Converting Data** dialob box closes unless errors were detected. If errors were detected, examine the errors listed in the **Errors** list box, then click the **Close** button. For additional details, go to *C:\Documents and Settings\All Users\Honeywell\Experion* and examine the ErrLog_*n*.txt error log file. The Export, Transform, and Import phases of the Conversion process are performed against all of the selected PIDs. For example, if selected PIDs are in CM1, CM2 and CM3; then Export is performed on CM1, CM2 and CM3; Transform is performed on CM1, CM2 and CM3; and Export is performed on CM1, CM2, and CM3. The audit trail of these phases appears in the ErrorLog_*n*.txt file. If the conversion does not complete successfully for any reason, a message appears in the log identifying which of the phases failed, with a reason for the failure. If either the Export or Transform phase fails. NONE of the selected PIDs will be converted. However, if the Import phase fails, some of the selected PIDs may be converted successfully. Completed conversions are listed in the log.

Step	Action
6	If error codes were listed, check the Control Builder Error Codes Reference to identify the numerical code. For example, if error message [EXPKS_E_CBC_COMPLETEWITHERRORS (1L.101.12477)] is listed, check for code 12477 in the document. Take the appropriate steps to correct errors. Errors would be those occurring during the Control Builder Import process.
7	Go to the next procedure *Configuration Phase* to configure the converted block.

Configuration phase

Prerequisites:

- Control Builder is running
- Tree windows are open

Considerations:

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

Complete the following steps to configure the new PID-PL block(s).

Step	Action
1	In the **Project** tab, double-click the converted CM icon to open its control drawing.
2	Double-click the new **PID-PL** block to access its configuration form. Modify all aspects of configuration as needed, beginning with the algorithm selection. Click the **Algorithm** tab and select **PROFITLOOP** for Control Equation Type.
	**ATTENTION** The equation for this block will not change to **PROFITLOOP** until you select it on the **Algorithm** tab of the PID-PL configuration form and you close the control drawing and save the changes.
3	If you want help defining the Profit Loop PKS model, use the **Profit Loop PKS Assistant**. To do so from the configuration form, click the **Advanced** tab, then the **Start Assistant** button. For details on using the Assistant, see *Profit Loop PKS Assistant overview*.

Step	Action
	**ATTENTION** You may receive a message that the block cannot be locked. Read this message and take one of the actions suggested. If accessing the Assistant from the Control Builder **Project** tree and block configuration form, you may have to answer **No** to the message to close the assistant and retrun to Control Builder. Close the Control Module and save changes, then repeat Steps 1 and 3. Once the block can be unlocked, the **Profit Loop PKS Assistant** will open.
4	After completing configuration, click **OK** to accept your changes and close the configuration form.

Download phase

After conversion and configuration, you can load the new PID-PL block to the controller. Refer to *Loading a control* strategy as needed.

Naming considerations

If the name of the **PID** block to be converted begins with "PIDA" (using capital letters), then you can retain the block name (PIDA), or change it to include a Profit Loop PKS designation as part of the conversion process. If you do not retain the name, then the name will change to include "_PLA" as part of the name. For example:

- PIDA will become PID_PLA.
- PIDA_1 will become PID_PLA_1.

If the **PID** block to be converted does not begin with "PIDA," then the original name will be used regardless of your conversion choice.

Conversion files

The conversion process creates various files, including:

- Configuration data files (.xml):
  - *CMName*.cnf.xml: The final file used for input to the Import function. This file contains the PID-PL block configuration data.
  - xxPID_*CMName*.cnf.xml: The original file created by the Export function. This file is unchanged, except for the name, and contains the original PID block configuration data.

- Conversion file (.sl):Used internally by the conversion function.

Reverting to a PID block

If you converted a **PID** block to a **PID-PL** block and wish to revert to the original **PID** block, complete the following steps.

Prerequisites:

- Control Builder is running
- Tree windows are open

Considerations:

- This procedure requires that you have not deleted the files created or moved during the block conversion.

Step	Action
1	In Control Builder, click **File** > **Import**.
2	In the **Import** dialog box, **Browse** to the directory used to store the original CM object. Be sure this directory does not contain the converted CM. The default location is *C:\Progrm Files\Honeywell\Experion\Engineering Tools\Ixport\export_buildX* folder.
3	In the **Object** list, click the name of the CM that contained the **PID** block that was converted. For example, if CM name was pidloop, select pidloop in the list.
4	Click the **Import** button. The **Importing Data** dialog box appears to track the progress of the import. When the import is completed, error message 14185 will appear in the Errors list on the dialog to let you know that the PID-PL block was deleted. Click the Close button to acknowledge the message and complete the import.

Profit Loop PKS Assistant overview

Defining a model

The Profit Loop PKS Assistant is a companion intended to simplify your Profit Loop PKS configuration activities. The configuration of Profit Loop PKS regulatory control point (PID-PL) does not require a good understanding of the math and concepts involved in model predictive control because the Profit Loop PKS Assistant will aid you in the development of the model. If you are uncertain about completing the Model

area in the **Advanced** tab of the PID-PL configuration form, the Profit Loop PKS Assistant provides several tools to help guide you through the model definition.

Since Profit Loop PKS is a model-based predictive controller, it must have a model. The Profit Loop PKS Assistant simplifies the model definition process by providing the following tools:

- **Model by Loop Type**: Default model parameter values can be specified by loop type and then modified for individual loops.
- **Model from PID Tuning**: The Profit Loop PKS model is calculated from existing PID tuning constants.
- **Model by Step Testing**: An automatic step generator and model identifier calculates the model from the step responses.
- **Model by Direct Entry**: When you have an existing LaPlace model from another source, you can enter the details directly.

Troubleshooting loop performance

The Profit Loop PKS Assistant provides diagnostic tools to assist with troubleshooting loop performance problems and performance improvement. These include:

- Fine Tuning: You can enter specific tuning parameters for a PID-PL loop, obtain current controller values, and download your settings to the controller.
- Valve Doctor: For flow and pressure loops , Profit Loop PKS Assistant analyzes valve stickiness probability. This analysis is automatically performed.
- Loop analysis: Provides an indication of loop health, based on oscillation detection, valve conditions, PV/OP standard deviations, and other factors.

Starting the Assistant

Three methods exist for accessing the Profit Loop PKS Assistant:

- Control Builder:
  - Starting from the Project tab
  - Starting from the Monitoring tab
- Experion:
- Starting from an Experion Station Point detail display

Starting from the Project tab

Prerequisites:

- Control Builder is running
- Tree windows are open

Follow these steps to start the Profit Loop PKS Assistant from within a Control Module (CM) drawing, from the Control Builder Project tab.

Step	Action
1	If you have not already completed these steps, do so now: • Start Control Builder. • On the **Project** tab, double-click the needed CM icon in the **Project** tab to open your control drawing. • If a PID-PL function block has not been added to the CM drawing, add one by dragging it from the Regulatory Control group in the **Library** tab to the Control Module drawing.
2	Double-click the **PID-PL** block to open its Parameter Configuration Form.
3	Click the **Advanced** tab.
4	Click the **Start Assistant** button. A status box displays details as the Profit Loop PKS Assistant starts, followed by a **Login Information** dialog box.
5	Complete the **Login Information** dialog box by entering your Control Builder user name and password, and click **OK**.
6	The **Profit Loop PKS Assistant** opens as a separate tool. Note that the configuration form remains open unless you close it. The **Profit Loop PKS Assistant** runs separately.
7	This completes the procedure.

Starting from the Monitoring tab

Prerequisites:

- Control Builder is running
- Tree windows are open

Follow these steps to start the Profit Loop PKS Assistant from within a Control Module (CM) drawing, from the Control Builder Monitoring tab. This procedure is applicable when the Control Module has been loaded to the C200 or ACE.

Step	Action
1	In the **Monitoring** tab, double-click the **PID-PL** block icon to open its Parameter Configuration Form.
2	Click the **Advanced** tab.
3	Click the **Start Assistant** button. A status box displays details as the Profit Loop PKS Assistant starts, followed by a **Login Information** dialog box.
4	Complete the **Login Information** dialog box by entering your Control Builder user name and password, and click **OK**.
5	The **Profit Loop PKS Assistant** opens as a separate tool. Note that the configuration form remains open unless you close it. The **Profit Loop PKS Assistant** runs separately.
6	This completes the procedure.

Starting from an Experion Station Point detail display

Prerequisites:

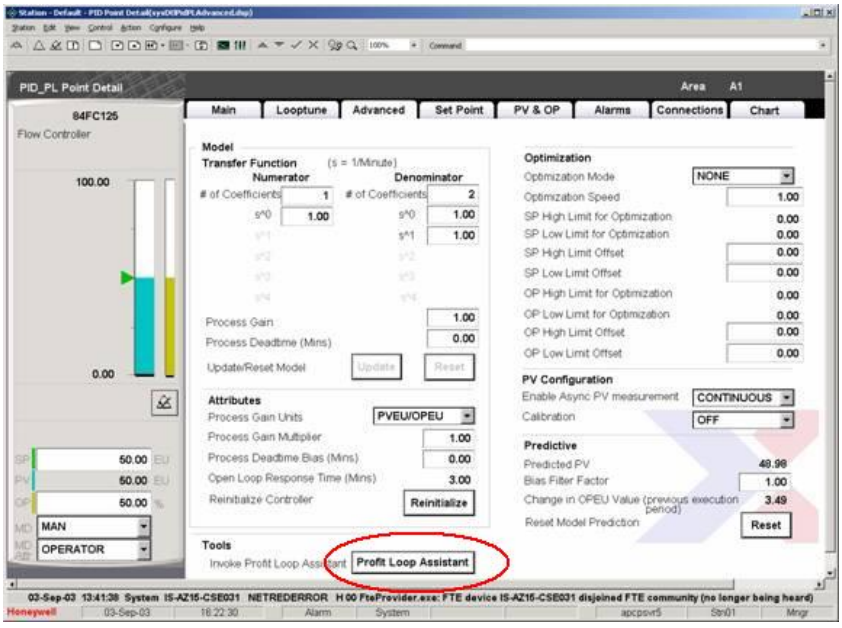
- Control Builder is running
- Tree windows are open

Considerations:

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

Follow these steps to start the Profit Loop PKS Assistant from the Station Point Detail display for a PID-PL point.

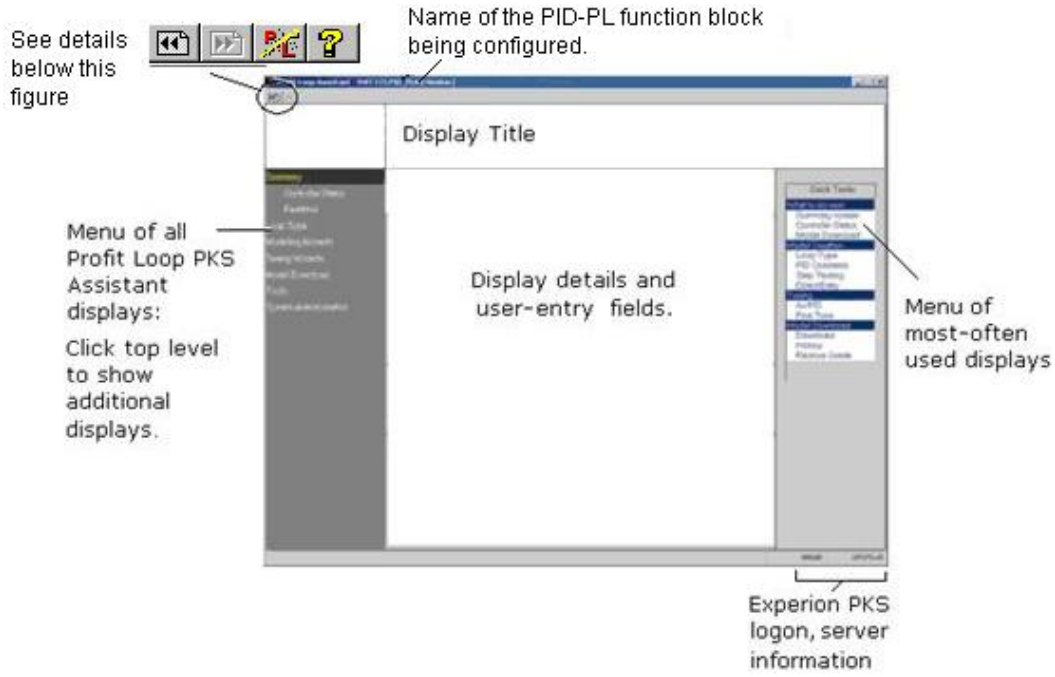
Step	Action
1	If you have not already called up the required **Point Detail** display, do so.
2	Click the **Advanced** tab.
3	In the **Tools** section, click the **Profit Loop Assistant** button.

Step	Action
	 A status box displays details as the **Profit Loop PKS Assistant** starts, followed by a **Login Information** dialog box.
4	Complete the Login Information dialog box by entering your Control Builder user name and password and click **OK**. The **Profit Loop PKS Assistant** opens as a separate tool. Note that the **Point Detail** display remains open unless you close it. The **Profit Loop PKS Assistant** runs separately.
5	This completes the procedure.

Working with Profit Loop PKS Assistant displays

General layout of displays

All Profit Loop PKS Assistant displays share common elements shown in the following figure.



Buttons in the upper left corner of every display include:

Button	Use
	Jump back to previous display, or forward to a display you recently visited.
	Display details about the Profit Loop PKS Assistant, such as file version and status information.
	Access the online Help, which contains details about every display.

Working with graphs and trends

You can enlarge a specific area of ("zoom in") nearly all of the trends or graphs presented in the Profit Loop PKS Assistant. For example, you can zoom in the Real-Time Trend display.

Prerequisites;

- Control Builder is running
- Tree windows are open

Step	Action
1	In a **Profit Loop PKS Assistant** display containing a trend or graph, identify the area of the graph you want enlarged.
2	Click and drag the cursor over the area to be enlarged. The selected area will become larger within the graph.
3	To return to the graph's original size, click the  button in the bottom left corner of the graph area

Obtaining basic information

Prerequisites:

- Control Builder is running
- Tree windows are open

Complete the following steps to obtain basic information about a PID-PL-based loop in an existing controller.

Step	Action
1	If the **PID-PL** function block of interest has not been selected in Control Builder or Experion Station display, select it.
2	If **the Profit Loop PKS Assistant** is not open, open it as explained in *Starting the Assistant*.
3	The Assistant opens to the Summary display. This display provides a synopsis of the selected PID-PL loop configuration. It also includes links to the displays where the settings can be modified and/or more details are available. See the **Profit Loop PKS Assistant** online Help for information about the contents of this (or any) display. Return to the Summary display from other displays by clicking **Summary** in the **Profit Loop PKS Assistant** "left menu" (the menu on the left side of the display).
4	To see other details: • For current controller settings and status, click **Summary** > **Controller Status** in the left menu. • For a trend view of the PV, SP, and OP, click **Summary** > **Real-Time Trend** in the left menu. • If you have defined character strings of tagnames for loop type identification, click **Loop Type** in the left menu to learn which loop type has been identified by the Assistant. See also *Phase 4: Define character strings for automatic* loop identification in this document.

Defining a model with Profit Loop PKS Assistant

The Profit Loop PKS Assistant includes four modeling wizards for creating the needed model.

To…	Use this procedure…
Set up an approximate model based on loop type.	Defining a Model by Loop Type
"Extract" a model from a working PID's tuning constants.	Defining a Model from PID Tuning
Use step-testing to identify an accurate model	Defining a Model by Step Testing

To…	Use this procedure…
Enter a model identified from another source	Define Model by Direct Entry

Define Model by Direct Entry

When you have an existing LaPlace model from another source, you can enter the details directly. Complete the following steps to define a model using the Model by Direct Entry.

Prerequisites:

- Control Builder is running
- Tree windows are open

Step	Action
1	Obtain the coefficient details of the LaPlace model you want to use.
2	If the **Profit Loop PKS Assistant** is not open, open it as explained in *Profit Loop PKS Assistant overview*.
3	If you want to compare the model you want to use with the one currently in your controller, go to step 4. If not, go to step 5.
4	To see the current controller model, click **Summary** in the **Profit Loop PKS Assistant** "left menu" (the menu on the left side of the display), then click Controller Status. Compare this model to the one you want to use.
5	To enter a model, click **Modeling Wizards** in the left menu.
6	In the list of wizards, click **Model by Direct Entry**.
7	In the **Direct Entry** display, enter the Laplace model coefficients. For help on specific entries on this display, refer to the **Profit Loop PKS Assistant** *online Help*.
8	If you are ready to load this model to the controller, click **Model Download** in the left menu.
9	In the **Model Downloads** display, select **Direct Entry** in the **To Download** list.
10	Click the **Download Settings** button.

Step	Action
11	To synchronize the Profit Loop PKS Assistant model definition with the Control Builder configuration, **upload** the newly loaded model from the controller to Control Builder.

Defining a Model from PID Tuning

This approach works especially well when you have converted from an existing **PID** function block with sound tuning constants. Even after conversion, the Assistant "remembers" the PID configuration (because a file containing the original PID data was saved). Thus, tuning constants from the original **PID** block can be used.

If your **PID-PL** function block was not created by conversion of a **PID** block, but you have tuning constant data you want to use, you can enter those values instead.

Prerequisites:

- Control Builder is running
- Tree windows are open

Complete the following steps to define a model using the Model from PID tuning.

Step	Action		
1	If you will be entering tuning values instead of recalling data from a **PID** function block, obtain the needed data.		
2	In Control Builder or the Experion Station display, select the PID-PL function block of interest.		
3	If the **Profit Loop PKS Assistant** is not open, open it as explained in *Starting the Assistant*.		
4	In the **Profit Loop PKS Assistant** "left menu" (the menu on the left side of the display), click **Modeling Wizards** > **Model from PID Tuning**.		
5	If you are retrieving existing **PID** values from an existing **PID**, skip to step 9. If you are entering tuning data, enter values for the following parameters:		
	• K	• T1	• T3
6	Select the PID's current equation type (Eqn Number).		
7	Select the PID's current control action.		

Step	Action
8	Enter a deadtime, in minutes.
9	If you are ready to load this model to the controller, click **Model Download** in the left menu.
10	In the **Model Downloads** display, To Download list, select Direct Entry.
11	Click the **Download Settings** button.
12	To synchronize the Profit Loop PKS Assistant model definition with the Control Builder configuration, **upload** the newly loaded model from the controller to Control Builder.

Defining a Model by Loop Type

If you know the dynamics of the loop, but do not want to perturb the process, the model can be selected from pre-defined loop dynamics. Complete the steps in the following phases to define a model by loop type:

- Phase 1: Phase 1: Preparation
- Phase 2: Phase 2: Establish area-group relationships
- Phase 3: Phase 3: Define loop dynamics by equipment number
- Phase 4: Phase 4: Define character strings for automatic loop identification
- Phase 5: Use Model by Loop Type

Phase 1: Preparation

The first phase in defining a model by loop type is preparation. This approach uses associations that you make between the areas in your plant (as defined in Experion) and groups and equipment types you identify in the Profit Loop PKS Assistant, to define models with similar loop dynamics.

It may help to "think backwards," by identifying the various controllers that have similar loop dynamics, then identifying ways that these can be grouped in terms of the areas in which they reside and the equipment with which they are associated.

Prerequisites:

- Control Builder is running
- Tree windows are open

Complete the following steps to prepare for model by loop type.

Step	Action
1	Decide on a group and equipment scheme based on the equipment in your plant's areas. This allows you to define models for equipment with similar loop dynamics across areas of your plant.
2	Document your scheme as needed for internal reference. You may find it helpful to use a Planning worksheet.
3	Continue with Phase 2: Establish area-group relationships.

Planning worksheet

If desired, use the following type of worksheet to set up your model by loop type scheme. For up to five groups, identify the areas to be associated with each group. For up to five equipment types, check the loop types to be defined.

Group # (1 - 5): _____					
Areas included:					**Equipment grouping or other details**
Equipment	**Loop type(s)**				
1		Flow		Analyzer	
		Level		Duty	
		Pressure		pH (External)	
		Temperature		Unknown	
2		Flow		Analyzer	
		Level		Duty	
		Pressure		pH (External)	
		Temperature		Unknown	
3		Flow		Analyzer	
		Level		Duty	
		Pressure		pH (External)	

Group # (1 - 5): _____					
Areas included:					**Equipment grouping or other details**
Equipment	**Loop type(s)**				
		Temperature		Unknown	
4		Flow		Analyzer	
		Level Pressure		Duty pH (External)	
		Temperature		Unknown	
5		Flow		Analyzer	
		Level		Duty	
		Pressure		pH (External0	
		Temperature		Unknown	

Phase 2: Establish area-group relationships

Phase 2 of the model-by1loop-type process is to associate Experion areas with Profit Loop PKS groups, based on your Phase 1: Preparation.

Prerequisites:

- Control Builder is running
- Phase steps 1 was completed

Complete the following steps to associate Experion with Profit Loop PKS groups.

Step	Action
1	If the **Profit Loop PKS Assistant** is not open, open it as explained in *Profit Loop PKS Assistant overview*.
2	In the **Profit Loop PKS As**sistant “left menu” (the menu on the left side of the display), click **System Administration** > **Model by Loop Type** > **Model Area Map**.
3	In the **Model Area Map** display, all areas defined in Experion are listed. Locate the first area you have identified for Model by Loop Type and click the

Step	Action
	group number with which the area is to be associated.
4	Repeat step 3 for all areas identified during Phase 1: Preparation. Refer to internal documentation as needed. The first twenty areas you map are considered Area Set 0. If you need to map more than 20 areas, go to step 5. Otherwise, go to step 7.
5	If you have more than 20 areas, in the **Area Set** field (at the top of the display), select **Area Set 1**. These sets are merely to allow **Profit Loop PKS Assistant** to manage and display all of your areas in the **Model Area Map** display. It is recommended that you document these sets in your internal documentation.
6	Continue mapping (steps 3 - 5) for the remainder of the areas, selecting a new Area Set when necessary.
7	Continue with *Phase 3: Define loop dynamics by* equipment number.

Phase 3: Define loop dynamics by equipment number

Phase 3 of the model-by1loop-type process is to establish particular loop dynamics for particular loop types, for equipment in the groups you have defined in Phase 2: Establish area-group relationships.

Prerequisites:

- Control Builder is running
- Phase steps 1 and 2 were completed

Complete the following steps to establish particular loop dynamics for particular loop types.

Step	Action
1	If the **Profit Loop PKS Assistant** is not open, open it as explained in *Starting the Assistant*.
2	In the **Profit Loop PKS Assistant** “left menu” (the menu on the left side of the display), click **System Administration** > **Model by Loop Type** > **Model Definition**.

Step	Action		
3	In the **Model Definitions** display, **Group Number** list, select the first group to be defined.		
4	In the **Equipment Number** list, select the first equipment to be defined, as determined during Phase 1: Preparation. Refer to internal documentation as needed.		
5	For each loop to be defined, enter loop dynamic values:		
	• Gain • Time Constant	• Normal DeadTime • Long DeadTime	• Short DeadTime
	Note the following: • It is recommended that you document these details in your internal documentation. • Leave the default values for loop types that are not relevant to the group-equipment you have selected. • If you have an existing text file exported from Profit Loop PKS at a previous time or on another server, you can import the data as explained in Exporting and importing Model Definitions.		
6	Repeat steps 4 and 5 for all equipment numbers within the group.		
7	Repeat steps 3 - 6 for all groups.		
8	Continue with *Phase 4: Define character strings for automatic* loop identification.		

Phase 4: Define character strings for automatic loop identification

Phase 4 of the model-by1loop-type process is optional, but can improve the efficiency of your configuration activities.

Prerequisites:

- Control Builder is running
- Phase steps 1 through 3 were completed

Considerations:

- Completing this phase allows you to automate loop identification for selected points. This is possible because you affiliate character strings, based on the point naming conventions at your plant, with particular loop types (also called "setting up Control Module name filters"). The Assistant will then identify the loop type of a selected point based on its name.

  Note that if you choose not to define these strings, or want a selection other than that made by the Assistant, you can select the loop type "manually."

Complete the following steps to define character strings for automatic loop identification.

Step	Action
1	Gather information about the point naming conventions at your plant, as they pertain to loops. For example, temperature-related points (which would be in a temperature loop) might have names that include the letters "TC," or analyzer-related points might include the letters "AN."
2	If the **Profit Loop PKS Assistant** is not open, open it as explained in *Starting the Assistant*.
3	In the **Profit Loop PKS Assistant** "left menu" (the menu on the left side of the display), click **System Administration** > **General Settings** > **Loop Type Filters**.
4	In the **Setup CM Name Filters for Loop Type** display, for a needed loop type, enter the character strings you identified in step 1 (up to five filters). Use wildcard characters as follows: • * (asterisk): Represents any number of characters. • ? (question mark): Represents a single character. For example, entering `*TC*` in the Temperature row causes **Profit Loop PKS Assistant** to identify points named "3TC100" and "5TC201" as temperature loop points.
5	Repeat steps 3 and 4 for all loop types relevant to your process control environment.
6	Click the Apply button to save the definitions.
7	Continue with *Use Model by Loop Type*.

Use Model by Loop Type

Once you have completed phases 1 - 4 of the model-by-loop-type process, Model by Loop Type must be chosen for a selected PID-PL function block, and the definition evaluated and modified if warranted by existing circumstances.

Prerequisites:

- Control Builder is running
- Phase steps 1 through 4 were completed

Complete the following steps to use Model by Loop Type for your model definition.

Step	Action		
1	In Control Builder or Experion Station display, select the **PID-PL** function block of interest.		
2	If the **Profit Loop PKS Assistant** is not open, open it as explained in *Starting the Assistant*.		
3	In the **Profit Loop PKS Assistant** “left menu” (the menu on the left side of the display), click **Loop Type**.		
4	In the **Loop Type** display, either verify the information or make the needed selections:		
	• Loop Type	• Integrator	• Is Valve
	Refer to the **Profit Loop PKS Assistant** *online Help* for details about these selections.		
5	Once the loop type details are entered or verified, in the left menu, select **Modeling Wizards** > **Model by Loop Type**.		
6	Evaluate all of the data on this display, and modify it if needed, to match your process conditions. The model details here are based on the entries you made in phase 3 for the selected loop type.		
7	If you make modifications, but want to start again from the default model, click **Reset To Default**. The values defined on the **Model Definition** display will be re-entered on this display.		
8	When you are satisfied with the model definition, proceed to *Download a model.*		

Defining a Model by Step Testing

Description

If you have no model data from other sources (such as PID tuning constants or an existing Laplace model), and it is possible to perturb the process, use step-testing to create a model.

Prerequisites:

Complete the steps in the following phases to define a model by step testing:

- Phase 1: Phase 1: Preparation
- Phase 2: Phase 2: Run the step test
- Phase 3: Phase 3: Evaluate model by step-testing results

Phase 1: Preparation

The first phase of defining a model by step testing is preparation. Complete the following steps to set up the conditions of the step test. This involves three procedures:

- Defining general settings for all step tests. See General step-testing settings.
- Defining settings specific to the PID-PL function block whose model you want to define. See Specific step-testing settings.
- Making advanced settings if desired. See Advanced step-testing settings.

General step-testing settings

Complete these steps to begin the set-up for Model by Step Testing.

Step	Action
1	If the **PID-PL** function block of interest has not been selected in Control Builder or Experion Station display, select it.
2	If the **Profit Loop PKS Assistant** is not open, open it as explained in *Starting the Assistant*.
3	In the **Profit Loop PKS Assistant** “left menu” (the menu on the left side of the display), click **System Administration** > **Model by Step-Testing**.

Step	Action		
4	For each loop type, enter the maximum number of minutes the step-test should run. Note that if the model parameters calculated by the identifier are still varying significantly, or the model quality is insufficient when this time is reached, the test will end.		
5	From the **Maximum Rank to Download** list, select a ranking from 1 through 5 to indicate the maximum quality of model you will allow to be downloaded to the controller:		
	• **1**: Excellent • **2**: Good	• **3**: Fair • **4**: Poor	• **5**: No model
	This rank is calculated by the model identification algorithm. If quality is not high enough (as determined by this selection), step testing will continue.		
6	If the loop type settings have not yet been made (or to change loop type), in the left menu, click Loop Type and go to the next step. Otherwise, skip to step 8.		
7	In the **Loop Type** display, make the following settings: • **Loop Type**: Verify or select the needed loop type. • **Integrator**: If the loop is an integrator, select Yes. If it is not an integrator, select No. This setting determines whether the OP (Open Loop) variable is stepped or the SP (Closed Loop) variable is stepped. • **Is Valve**: If the loop is directly connected to the valve, select Yes. If this is a primary loop in a cascade scheme, select No.		
8	Continue with *Specific step-testing settings*.		

Specific step-testing settings

Complete these steps to continue the set-up for Model by Step Testing.

Step	Action
1	In the **Profit Loop PKS Assistant** left menu, click **Modeling** Wizards > **Model by Step-Testing** > **Test Setup**.
2	In the **Model by Step-Testing Test Setup** display, review the loop information (Loop Info) to ensure it is correct. Note: • If the Loop Info is not correct, click the **Loop Type** link and make needed

Step	Action
	changes on the **Loop Type** display. • If you click the link, click the **Back** button,  , to return to the **Test Setup** display.
3	In the **Test Setup** display, review the Controller State information. Messages in this part of the display will alert you to errors. Note: • For an integrator, the controller Mode must be Auto. • For a controller that is not an integrator, the Mode must be Manual. • In both cases, the Mode Attribute must be Program. • If the controller mode and/or attribute are incorrect, click either of the corresponding links to access the Controller Status display. On this display, make the needed changes. • If you make these changes, click the **Back** button,  , to return to the **Test Setup** display.
4	In the Test Type area: • From the **Test Type** list, select either **PRBS** or **Manual**. • **Estimated Response Time**: Enter an estimation of the response time in minutes. • If **Test Type** is **PRBS**, enter an **Upper Bound** and **Lower Bound** for the step test.
5	Optionally, continue with *Advanced step-testing settings*. If you do not need to define any advanced settings, then go to *Phase 2: Run the* step test.

Advanced step-testing settings

If desired, you can make advanced settings for Model by Step Testing. Complete these steps to define advanced options, thereby deviating from the Profit Loop PKS Assistant default values.

Step	Action
1	On the **Model by Step-Testing Test Setup** display, click the **Advanced Settings** button on the right side of the display (under the **Quick Tasks** list). Or, in the left menu, click **Modeling Wizards** > **Model by Step-Testing** > **Advanced**.

<table>
<tr><th>Step</th><th colspan="2">Action</th></tr>
<tr><td>2</td><td colspan="2">On the Model by Step-Testing Advanced display, change any of the following items:</td></tr>
<tr><td></td><td>Step Test Settings</td><td>Model Type Settings</td></tr>
<tr><td></td><td>• Execution Frequency
• Move Filter
• Maximum Testing Time</td><td>• Desired Rank
• Dead Time Search Method
• Model Order</td></tr>
<tr><td></td><td colspan="2">Refer to the Profit Loop PKS Assistant online Help for details about these items.</td></tr>
<tr><td>3</td><td colspan="2">If you to return to the Profit Loop PKS Assistant defaults, click the Restore Defaults button. Otherwise, go to Phase 2: Run the step test.</td></tr>
</table>

Phase 2: Run the step test

Phase 2 of the model-by-step-testing process is to run the test, which will be conducted based on your step-test Phase 1: Preparation. Complete the following steps to run the test.

Step	Action
1	If the **PID-PL** function block of interest has not been selected in Control Builder or Experion Station display, select it.
2	If the **Profit Loop PKS Assistant** is not open, open it as explained in *Starting the Assistant*.
3	In the **Profit Loop PKS Assistant** “left menu” (the menu on the left side of the display), click **Modeling Wizards** > **Model by Step-Testing** > **Running**.
4	At the top of the **Running** display, select the graph details: • Current PV, SP, OP or Future OP or SP. • Timeframe of the graph. Refer to the **Profit Loop PKS Assistant** *online Help* for details about these selections.
5	If desired, change **Estimated Response Time**, and, if applicable **Upper** and/or **Lower Bound** values. Otherwise, go to step 6. These values were originally set on the **Model By Step-Testing Test Setup**

Step	Action
	display, but may be changed here. (If you do this, the **Test Setup** display will be updated to match.) Refer to the **Profit Loop PKS Assistant** *online Help* for details about these entries.
6	When all settings are complete, click **Start Test** button.
7	Monitor the step-test from the **Step-Test Ru**nning or **Results** display. Note the status messages on **Running** display. If needed, you can pause the test/resume the test, or stop the test.
8	There are several ways the test may end: • If the model quality is sufficient (as defined on the **System Administration** > **Model by Step Testing** display) and a number of identifications indicate the test is not changing much, the test will stop and the Assistant will automatically bring up the **Model Download** display. • If the model quality is insufficient or the test is changing significantly, and the maximum test time is reached, the test will stop. • You can stop the test at any time by clicking the **Stop** button. A positive reason to do this would be when a sufficient number of steps have been performed, the model quality is 3 or less, and the predictability value is 0.5 or greater.
9	When a model is identified during the test, you can view results as explained in *Phase 3: Evaluate model by* step-testing results.
10	Once an acceptable model is obtained, go to the **Model Downloads** display to download the model. See *Download a model* in this section for details.

Phase 3: Evaluate model by step-testing results

Phase 3 of the model-by-step-testing process is to evaluate the test results. Complete the following steps to evaluate the step-test. This procedure assumes that the test has just completed running.

Step	Action
1	In the **Model by Step-Testing Running** display, click the **Show Results** button. Optionally, in the left menu, click **Modeling Wizards > Model by Step-Testing** > **Results**.

Step	Action
2	Evaluate all details in the **Model by Step-Testing Results** display. A rank (CalcRank) of 3 or less, and a Predictability value of 0.5 or greater, is generally considered a fairly good model. For other details on the display contents, see the **Profit Loop PKS Assistant** online *Help*.
3	If you are satisfied with the model, download it by clicking the **Download** Model button, to access the **Model Download** display. Complete the process as explained in *Download a model*

Download a model

Profit Loop PKS Assistant within Experion

When you download a model from the Profit Loop PKS Assistant, you are downloading to the controller. As with other types of changes to the controller, you can upload the controller configuration to store the changes in Control Builder's Monitor-side Engineering Repository Database (ERDB), and then perform an update to store the changes in the Project-side ERDB.



REFERENCE - INTERNAL

For details on downloading and uploading, see *Loading a control* strategy in this document.

Prerequisites:

- Control Builder is running
- Tree windows are open

Considerations:

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

Downloading procedure

Complete the following steps to download a model definition from Profit Loop PKS to the controller.

Step	Action	
1	Define a model using one of these methods:	
	• Define Model by Direct Entry • Defining a Model from PID Tuning	• Defining a Model by Loop Type • Defining a Model by Step Testing
2	When you are satisfied with the model, click **Model Download** in the **Profit Loop PKS Assistant** "left menu" (the menu on the left side of the display). Ensure that you know which type of model it is (refer to step 1).	
3	In the **To Download** list, select the type of model to be downloaded.	
4	Review the model information.	
5	If desired, • Compare the model selected in step 3 to the model currently in the controller. • Compare all model definitions in the Assistant to the model in the controller.	
6	If the model to be downloaded is satisfactory, click the **Download Settings** button.	
7	In response to the confirmation message, click **Yes** to download the new model, or **Cancel** if you do not want to download the model.	

Reverting to a previous model

Description

You can take different actions, depending on the situation, to revert to a previously downloaded or defined model:

- Retrieve one of the past ten models downloaded.
- *Restore grades*. Note that if you converted a PID block to a PID-PL block and do not like the results, you can retrieve the PID block configuration; see Reverting to a PID block.

Prerequisites:

- Control Builder is running

- Tree windows are open

Retrieve historical model

Complete the following steps to retrieve one of the past ten models downloaded to a controller.

Step	Action
1	If the **Profit Loop PKS Assistant** is not open, open it as explained in *Starting the Assistant*.
2	In the **Profit Loop PKS Assistant** "left menu" (the menu on the left side of the display), click **Model Download** > **History**.
3	In the list of downloaded models, select the desired model.
4	Click the **Load to Wizard** button. The model will be imported into the **Profit Loop PKS Assistant** and the details will be added to the corresponding modeling wizard.
5	Confirm the model details presented at the bottom of the display. If needed, go to the corresponding **Modeling Wizard** display and make modifications.
6	When the model is acceptable, it can be downloaded to a controller as explained in *Download a model*.

Create grades

You can save any of the models currently configured in the Profit Loop PKS Assistant, or running in a controller, and associate them with grades of product (or particular products). This can be useful if you produce different grades (or types) of product, each requiring a different model. Each "grade" is a complete model definition. For details on this feature, refer to the Profit Loop PKS Assistant online Help.

Complete the following steps to create a set of grades.

Step	Action
1	If the **Profit Loop PKS Assistant** is not open, open it as explained in *Starting the Assistant*.
2	Use the Modeling Wizards to define a model(s) that you want to save.
3	If you want to save the model definition from a particular controller, select the **PID-PL** function block of interest in Control Builder or Experion Station

Step	Action
	display.
2	In the **Profit Loop PKS Assistant** “left menu” (the menu on the left side of the display), click **Model Download** > **Grades** > **Creating**.
3	In the **Grades** list in the middle of the display (*not* the **Grades Available** table), select the number of the next available grade. For the first grade, select 1.
4	Enter a short name and a description (up to 100 characters) for the grade.
5	In the **Current Models** list, select the model type you want to associate with this grade. To choose the model from the controller, select **Current Model**.
6	Review the model summary at the bottom of the display. If needed, go to the appropriate **Modeling Wizards** display and refine the model. In this case, you will need to begin this procedure after you are finished.
7	Click **Save Grade** to save the definition and add it to the **Grades Available** table.
8	Repeat this procedure for all models to be saved.

Restore grades

Complete the following steps to restore a model definition that has been saved as a grade.

Step	Action
1	If the **Profit Loop PKS Assistant** is not open, open it as explained in *Starting the Assistant*.
2	In the **Profit Loop PKS As**sistant “left menu” (the menu on the left side of the display), click **Model Download** > **Grades** > **Restoring**.
3	In the **Grades Available** list, select the grade you want to restore. Details of the selected grade display in the bottom of the display.
4	Review the grade details to ensure they match the model definition you are seeking.
5	Click the **Model to Wizard** button. The model will be imported into the **Profit Loop PKS Assistant** and the details will be added to the corresponding modeling wizard.

Modify grade name or description

Complete the following steps to revise a grade name or description.

Step	Action
1	If the **Profit Loop PKS Assistant** is not open, open it as explained in Starting the Assistant.
2	In the **Profit Loop PKS Assistant** "left menu" (the menu on the left side of the display), click **Model Download** > **Grades** > **Restoring**.
3	In the **Grades Available** list, select the grade whose name and/or description are to be revised.
4	Click the **Edit Name/Desc** button.
5	In the **Name** and/or **Description** fields, make needed changes.
6	Click the **Save Name/Desc** button.

Exporting and importing Model Definitions

Description

If you have many controllers in different areas of your plant, and similar tuning constants can be used across these areas, it may simplify your work effort to export definitions and then import them on various servers.

Prerequisites:

- Control Builder is running
- Tree windows are open

Export model definitions

Complete the following steps to export model definitions.

Step	Action
1	In **Profit Loop PKS Assistant**, select **System Administration** > **Model Definitions** from the left menu.

Step	Action
2	If you have not yet defined a set of definitions for each loop type, do so as explained in *Defining a Model by Loop* Type, particularly *Phase 3: Define loop dynamics by* equipment number.
3	On the **Model Definitions** display, click **Export**.
4	In the resulting **Export** dialog box, select the group numbers to export, then click Choose File.
5	In the resulting dialog box, select the drive and folder location to which the file should be saved, enter the file name, then click **Save File**. The selected group definitions will be saved to the file specified, and the extension ".ini" will be appended to the file.

Import model definitions

If you have exported model definitions as explained in Exporting and importing Model Definitions, complete the following steps to import them into Profit Loop PKS.

Step	Action
1	In **Profit Loop PKS Assistant**, select **System Administration** > **Model Definitions** from the left menu.
2	Identify the location of the model definitions file (a text file with an extension of ".ini") from another source. If needed, see *Export model definitions*.
3	In the left menu, select **System Administration** > **Model Definitions**.
4	On the **Model Definitions** display, click **Import**.
5	In the resulting **Import** dialog box, select the needed drive, folder, and file, then click **Open File**.
6	The group and equipment numbers defined in the file will display. Select the needed group and equipment numbers to be imported, then click **Import**. The definitions will be imported into this instance of **Profit Loop PKS Assistant**.

Copying model definitions

Prerequisites:

- Control Builder is running
- Tree windows are open

Complete the following steps to copy model definitions.

Step	Action
1	In **Profit Loop PKS Assistant**, select **System Administration > Model Definitions** from the left menu.
2	If you have not yet defined a set of definitions for each loop type, do so as explained in *Defining a Model by Loop* Type, particularly *Phase 3: Define loop dynamics by* equipment number.
3	Select a group and equipment number to be copied, click the **Copy** button.
4	In the resulting **Copy** dialog box, note that your selection is entered in the **Source** area.
5	Review the options in this dialog box.
6	To copy a group to multiple destinations, click the **Copy Entire Group** option.
	**ATTENTION** Selecting this option, causes the values for the equipment numbers in the group number selected to be copied to every equipment number for each group number selected as a Destination.
7	If the source equipment and group number values need to be copied to specific equipment numbers in specific groups: a) Select the **Copy Entire Group** option. b) Select the **Destination Group** numbers. c) Select the **Copy Only Equip** option. d) Select the **Destination** equipment numbers.
8	When the copy selections for source and destination have been made, click **Copy**.

Tune a PID-PL-based controller

Description

The Profit Loop PKS Assistant includes two tuning wizards for adjusting the tunable parameters of the controller:

- Tune as PID
- Fine Tune a controller

In addition, you can read valve-related guidelines as explained in *Obtain valve details*.

Prerequisites:

- Control Builder is running
- Tree windows are open

Tune as PID

You can enter PID tuning constants, which will be automatically converted to PID-PL model coefficients. Use this method to determine the model for your controller by tuning it as a PID loop if you are more comfortable with PID tuning constants.

Complete the following steps to enter PID tuning constants.

Step	Action
1	If the **PID-PL** function block of interest has not been selected in Control Builder or Experion Station display, select it.
2	If the **Profit Loop PKS Assistant** is not open, open it as explained in *Starting the Assistant*.
3	In the **Profit Loop PKS Assistant** "left menu" (the menu on the left side of the display), click **Tuning Wizards** > **Tune as PID**.
4	Adjust Gain, Integral Time, and Derivative Time values as needed. The Assistant automatically converts the data to the Laplace model needed by the controller.
5	Make other settings on the display as needed. Refer to **the Profit Loop PKS Assistant's** *online Help* for display details.
6	Evaluate the graph on the display to assess your entries.
7	If further adjustment is needed, see *Fine Tune a controller* in this section.

Step	Action
8	When satisfied with your entries, click the **Download Model** button to download the new settings to the controller. In response to the resulting confirmation message, click **Yes** to download the model.

Fine Tune a controller

You can enter specific tuning parameters for a PID-PL loop to assist in refining your model.

Complete the following steps to fine tune a controller

Step	Action
1	If the **PID-PL** function block of interest has not been selected in Control Builder or Experion Station display, select it.
2	If the **Profit Loop PKS Assistant** is not open, open it as explained in *Starting the Assistant*.
3	In the **Profit Loop PKS Assistant** "left menu" (the menu on the left side of the display), click **Tuning Wizards** > **Fine Tune**.
4	Make settings as needed. Refer to the **Profit Loop PKS Assistant's** *online Help* for display details.
5	If desired, you can retrieve the controller settings after you have made changes in this display. To do so, click the **From Controller** button.
6	When satisfied with your entries, click the **Download Settings** button to download the new settings to the controller.
7	In response to the confirmation message, click **Yes** to download the settings or **No** to cancel.

Obtain valve details

For suggestions on dealing with valves, access the Valve Problems display.

Step	Action
1	If the **PID-PL** function block of interest has not been selected in Control Builder or Experion Station display, select it.
2	If the **Profit Loop PKS Assistant** is not open, open it as explained in *Starting the Assistant*.
3	In the **Profit Loop PKS Assistant** "left menu" (the menu on the left side of the display), click **Tuning Wizards** > **Tune - Valve Problem**.
4	Read the suggestions and take corrective action as needed.

Troubleshooting a PID-PL-based controller

Description

If you have a controller configured with Profit Loop PKS, and it is performing poorly, you can use the Profit Loop PKS Assistant to troubleshoot the controller. Tools available include:

- Valve Doctor
- Loop Analysis

Prerequisites:

- Control Builder is running
- Tree windows are open

Use the Valve Doctor

The Valve Doctor uses online Experion history if it is available, or collects its own data from the controller while the Profit Loop Assistant is open with the controller selected. These data are used to determine whether or not the valve is sticky.

Step	Action
1	If the **Profit Loop PKS Assistant** is not open, open it as explained in *Starting the Assistant*.
2	In the **Profit Loop PKS Assistant** "left menu" (the menu on the left side of the display), click **Tools** to open the **Tools group**.
3	If the controller is a flow or pressure loop with outputs to a valve, and is being operated in a closed loop, click **Valve Doctor** in the **Tools** group (this menu item will not be available unless the proper type of loop is identified). The Valve Doctor analysis begins when you start the Assistant. The analysis is active whenever the button on this display is labeled "**Stop Analysis**." If the analysis is stopped, click **Restart Analysis** to start again.
4	Check the More Data entry. If it is 1, more data is needed before the analysis will be valid. To perform analysis, the Valve Doctor needs 1 hour of fast-sampled (faster than 5 seconds) PV, OP, and SP closed-loop operational data. This data may or may not exist in the server history. If the entry is 1, check the More Data Left entry to determine how many more minutes of data are needed.
5	If more data is not needed, the More Data entry will be 0. See the **Profit Loop PKS Assistant** *online Help* for more information on the More Data entries.
6	When sufficient, good data is indicated, evaluate the Valve Doctor's evaluation of valve stickiness: • **Sticky Probability**: High, medium, or low (or unknown) probability that valve stiction exists. • **OP Sticky**: Indicates how sticky a valve is. Entry is in the same units as the OP. For example, if the OP is a percentage between 0 and 100, then this number is normally between 0.3% and 2% for a typical sticky valve. • **PV Sticky**: PV stickiness caused by valve stiction, in the same units as the PV.
7	If the probability is High, and the OP and/or PV stickiness readings indicate a

Step	Action
	problem, check the valve and take corrective action as needed. For general guidelines on dealing with valves, see *Obtain valve details* in this document.

Use Loop Analysis

Loop Analysis provides an indication of loop health. The analysis consists of calculations for the following:

- Oscillation detection
- Valve-sizing problem detection
- PV resolution detection

Loop Analysis uses the best available existing data for the different calculations. The analysis is automatically performed when the Profit Loop PKS Assistant is started and the loop is determined to send its output to a valve.

Complete the following steps to analyze a loop.

Step	Action
1	If the **Profit Loop PKS Assistant** is not open, open it as explained in *Starting the Assistant*.
2	In the **Profit Loop PKS Assistant** "left menu" (the menu on the left side of the display), click **Tools** > **Loop Analysis**.
3	Evaluate the display contents, referring to the **Profit Loop PKS Assistant** *online Help* for details about the entries. Note that for this release, the Valve Travel Calculation and PV OP Variance Calculation are not used
4	If problems are identified by the Loop Analysis, make corrections as needed.

Controlling access to Profit Loop PKS

Description

Security, based on Experion role, can be configured to control access to Profit Loop PKS displays. Access can be allowed or denied for roles other than the administrative role, on per-display basis.

Prerequisites:

- Control Builder is running
- Tree windows are open

Complete the following steps to set access.

Step	Action
1	If the **Profit Loop PKS Assistant** is not open, open it as explained in *Starting the Assistant*.
2	In the **Profit Loop PKS Assistant** "left menu" (the menu on the left side of the display), click **System Administration** > **General Settings** > **Security**.
3	For each of the Assistant's displays, enter the required security level (Experion role), and whether or not access is allowed. Or, if desired, select a display, then make your settings in the **Security for Selected Tool** area of the display.
4	When finished, click the **Apply** button.

Loading a control strategy

This section includes information that you can reference for:

Topic	Link
Overview	Click here
Loading control stratgey components	Click here
Using Upload command	Click here
Using Upload With Contents command	Click here
Using Compare Parameters	Click here
Copying control strategies using templates	Click here
Loading only server configuration	Click here

About load operations

The system provides the ability to build control strategies offline, without being connected to the actual field components. The process of transferring the Control Strategy to the "live" working components in the field is called the load operation.

The load operation functionally copies configuration data from the control strategy that is stored in the Engineering Repository Database (ERDB) to the assigned field component in the system architecture. It indirectly assures that the planned system matches the actual one. The communication addresses and physical location assignments specified for components through Control Builder configuration must match the actual addresses and locations of components in the system.

Loaded versus project database versions

The master control strategy stored in the Engineering Repository Database (ERDB) is configured and edited through the Project tree. Once the contents of the control strategy are loaded from Project to the applicable components a loaded version of the Project or master database is created. The loaded version of the database is only viewable through the Monitoring tree and only supports minimal editing of the Control Strategy configuration data.

The following commands are included in the Control Builder Tools menu to synchronize data in the loaded database with the data in the Project/master database.

- Update to Project
- Update with Contents (to Project)

See the on-line help for information about these commands.

Load initiation and load dialog box

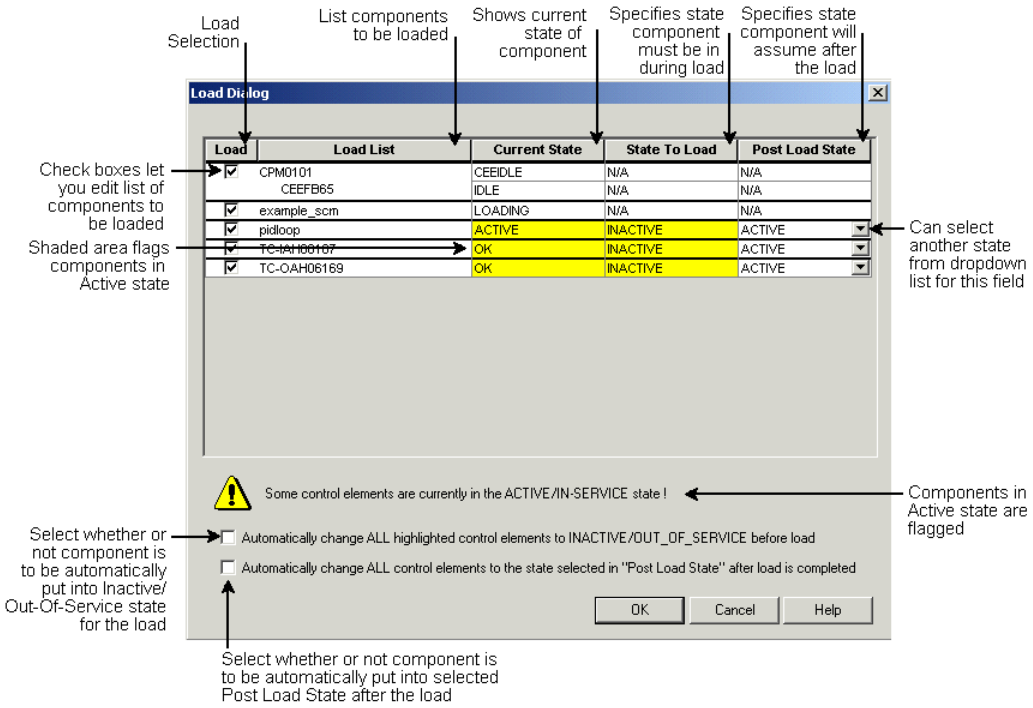
You can initiate a load function for selected components from either the Project tree or Monitoring tree using one of the following commands in the Control Builder Tools menu.

- Load
- Load with Contents

Either command invokes the Load Dialog box. The following figure shows a sample Load Dialog box invoked for a load with contents operation for a CPM. It provides a brief description of the dialog box features for quick reference. The appearance of the dialog box will vary depending on the current load circuminstances such as whether this is an initial load or a re-load operation.

CAUTION

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



Load action with compare parameters function

The capability of the load action is expanded when the Use Compare Parameters function is enabled through the System Preferences dialog. Please refer to the following Using Compare Parameters section for more information.

Load options for server history and server displays configuration

You can enable or disable the loading of history, trend, or group configuration data for a block to Server through the System Preferences dialog.
Please refer to the Setting system preferences section for more information.

Initial load order guidelines

Make the initial load of control strategy components from the Project tab in the following order to minimize possible load interaction generated error messages. Use the Load rather than the Load with Contents command.

Order	Component
1	Control environment components such as: CPM/CEE ACE/CEE SCE/CEE
2	Input/Output interface components such as: FIM/LINK IOLIM/IOLINK SIMIOLIM/SIMIOLINK
3	Redundancy related components such as: CPM (secondary) IOLIM (secondary) RM (primary and secondary)
4	Fieldbus devices
5	Input/Output Processors (IOPs)
6	Input/Output Modules (IOMs)
7	Control Modules (CMs) and/or Sequential Control Modules (SCMs)

Component deletion considerations

Control Strategy edits must be performed from the Project Tree only.

- Deleting blocks from the Project Tree eliminates them from the Project version of the database only. Only blocks that are not loaded can be deleted. Delete loaded blocks from the Monitoring Tree first before deleting them from the Project Tree.
- Deleting blocks from the Monitoring Tree eliminates them from the controller, Server and loaded version of the database. The blocks remain in the Project version of the database.



ATTENTION

Changes to parameters in the controller can be made from the Monitoring Tree. See Changing Parameters while Monitoring in the *On-line Monitoring using Control Builder* section.

Loading control strategy components

Loading CPM/CEE

Use the following general procedure to load a Control Processor Module and its associated Control Execution Environment (CEE). The load procedure is similar for all control environment related components.

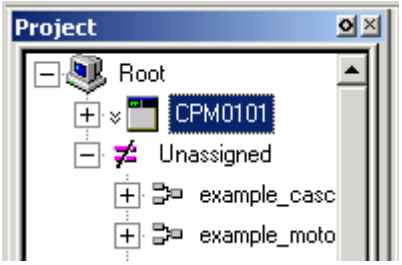
Prerequisites:

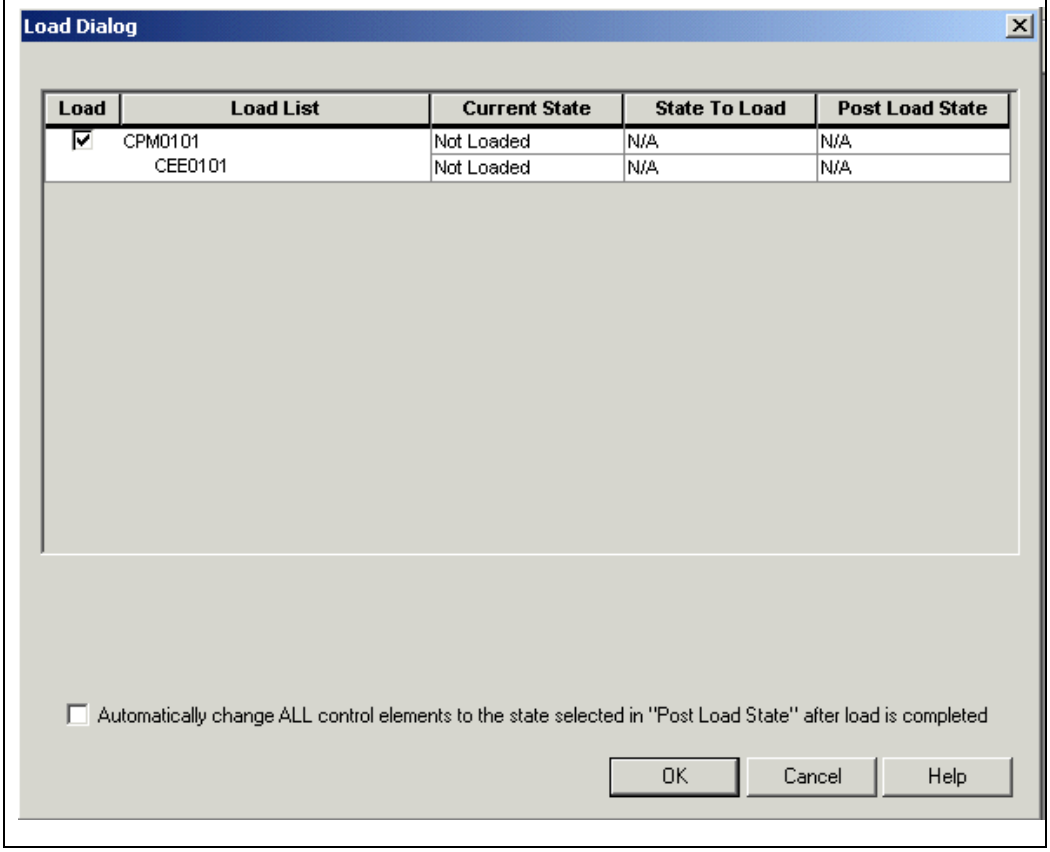
- Control Builder is running
- This procedure assumes that the Control Processor Module is installed and capable of communicating with the Server.

Considerations:

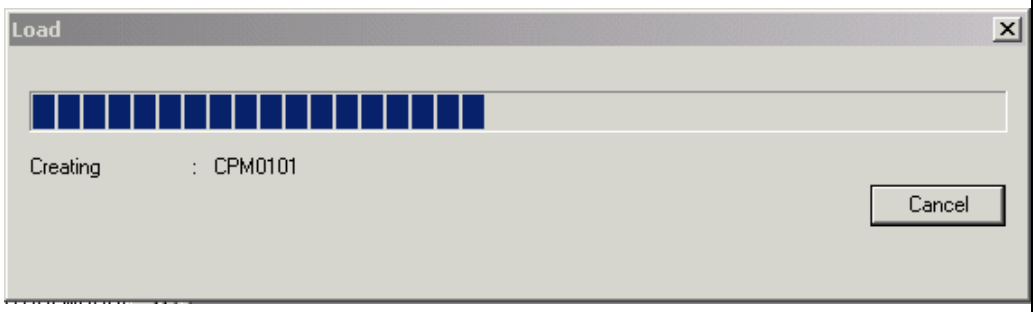
- All illustrations used in the procedure are for example purposes only.

Step	Action	Result
1	Click desired CPM block icon in Project tab.	Selects and highlights the component.

Step	Action	Result
		
2	Click Tools->Load. Or, click the  load button in the toolbar.	Calls up Load Dialog box



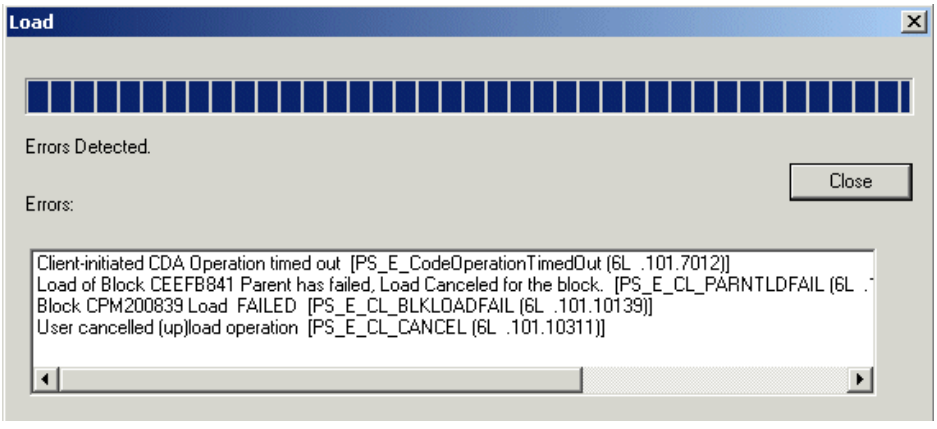
Step	Action	Result
3	With load check box checked, click the OK button.	Initiates the load to the CPM and calls up the load progress dialog.



TIP

You can also check load progress through the four-LED display on the front panel of the Control Processor Module. The display changes from NODB to NOEE to IDLE upon a successful Load.

If errors are detected, they will be displayed in the Load progress dialog and you will be asked if you want to continue the load or cancel, depending on the nature of the error. We suggest that you cancel the load and identify and fix the errors. The following illustration shows how error messages are typically displayed. Each message includes an error code in parentheses. Note the last number in the string. In some cases, more information about the code number may be included in the Control Builder Notifications Reference document in Knowledge Builder.



Step	Action	Result
4	Once the load completes and the dialog box closes, click the Monitoring tab.	CPM/CEE icons now appear in Monitoring tab. The default state for a loaded CPM/CEE is inactive/idle or color code blue.
5	See the Activating the CEE section in this document to set the CPM to its Run state.	CPM/CEE icons turn green when active.
6	Repeat this procedure for other control components as required.	

Loading IOLIM/IOLINK

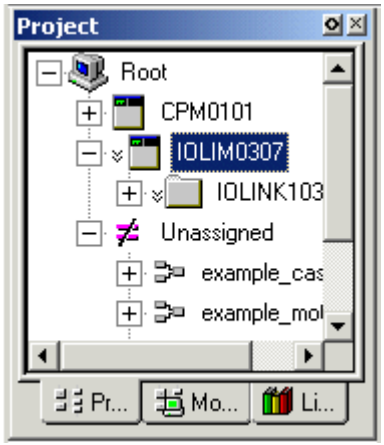
Use the following general procedure to load an I/O Link Interface Module (IOLIM) and its associated I/O Link (IOLINK). The load procedure is similar for all I/O interface related components.

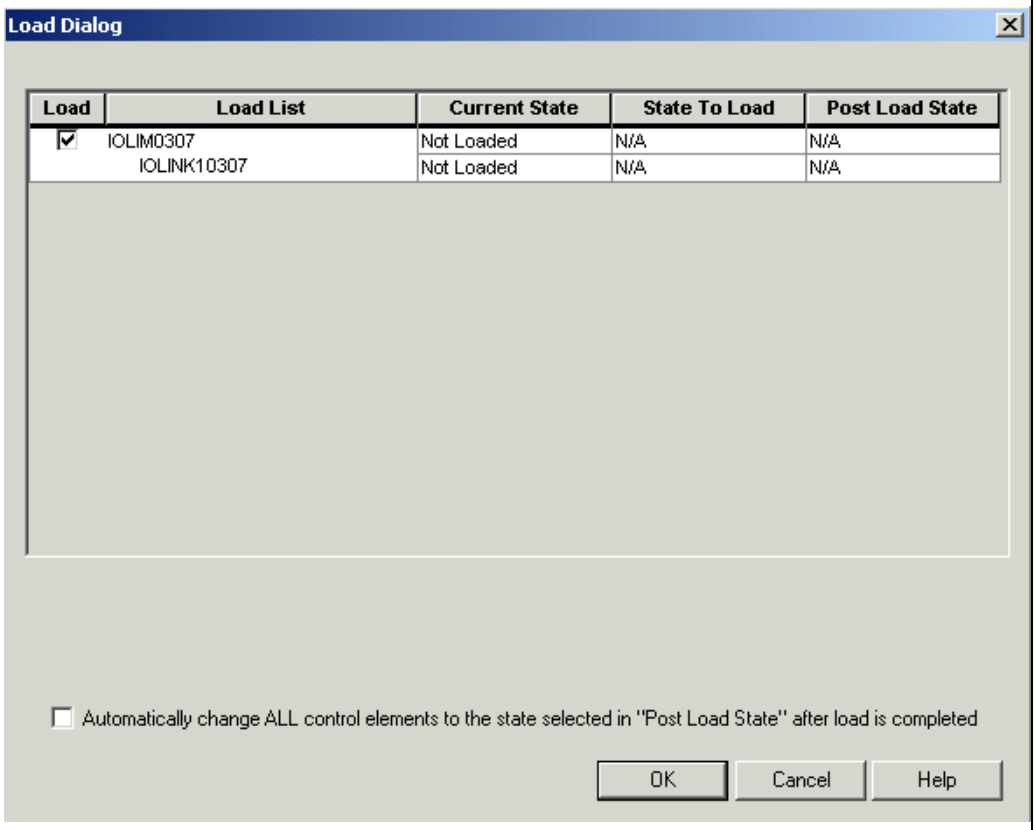
Prerequisites:

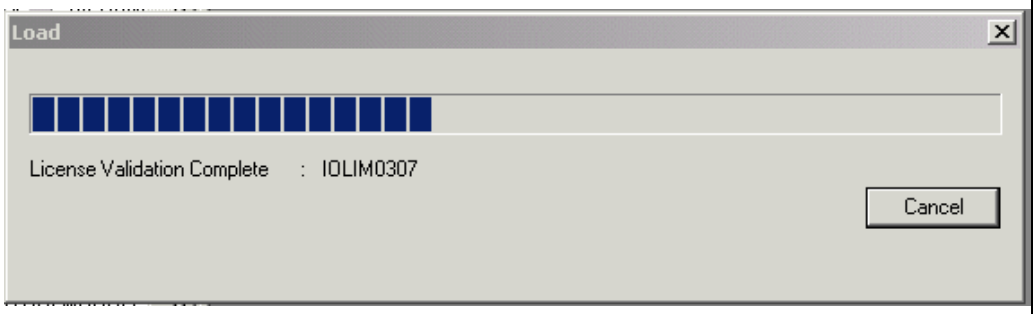
- Control Builder is running
- This procedure assumes that the IOLIM is installed and capable of communicating with the Server.

Considerations:

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

Step	Action	Result
1	Click desired IOLIM block icon in Project tab.	Selects and highlights the component. 
2	Click Tools->Load. Or, click the  load button in the toolbar.	Calls up Load Dialog box

Step	Action	Result
		
3	With load check box checked, click the OK button.	Initiates the load to the IOLIM and calls up the load progress dialog.

Step	Action	Result
		
	TIP You can also check load progress through the four-LED display on the front panel of the I/O Link Interface Module. The display changes from NODB to NOEE to OK upon a successful Load. If errors are detected, they will be displayed in the Load progress dialog and you will be asked if you want to continue the load or cancel, depending on the nature of the error. We suggest that you cancel the load and identify and fix the errors. Each message includes an error code in parentheses. Note the last number in the string. In some cases, more information about the code number may be included in the Control Builder Notifications Reference document in Knowledge Builder.	
4	Once the load completes and the dialog box closes, click the Monitoring tab.	IOLIM/IOLINK icons now appear in Monitoring tab. The default state for a loaded IOLIM/IOLINK is active or color code green.
5	Repeat this procedure for other I/O interface components as required.	

Loading RM

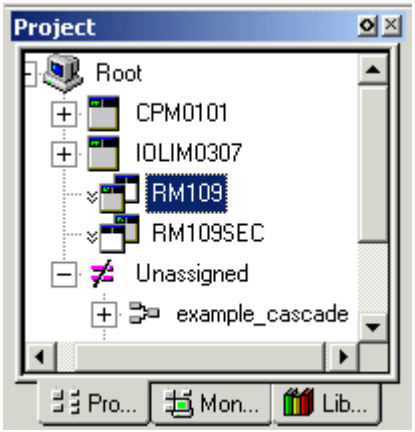
Use the following general procedure to load a Redundancy Module (RM) and its associated secondary Redundancy Module. The load procedure is similar for all primary and secondary redundant components.

Prerequisites:

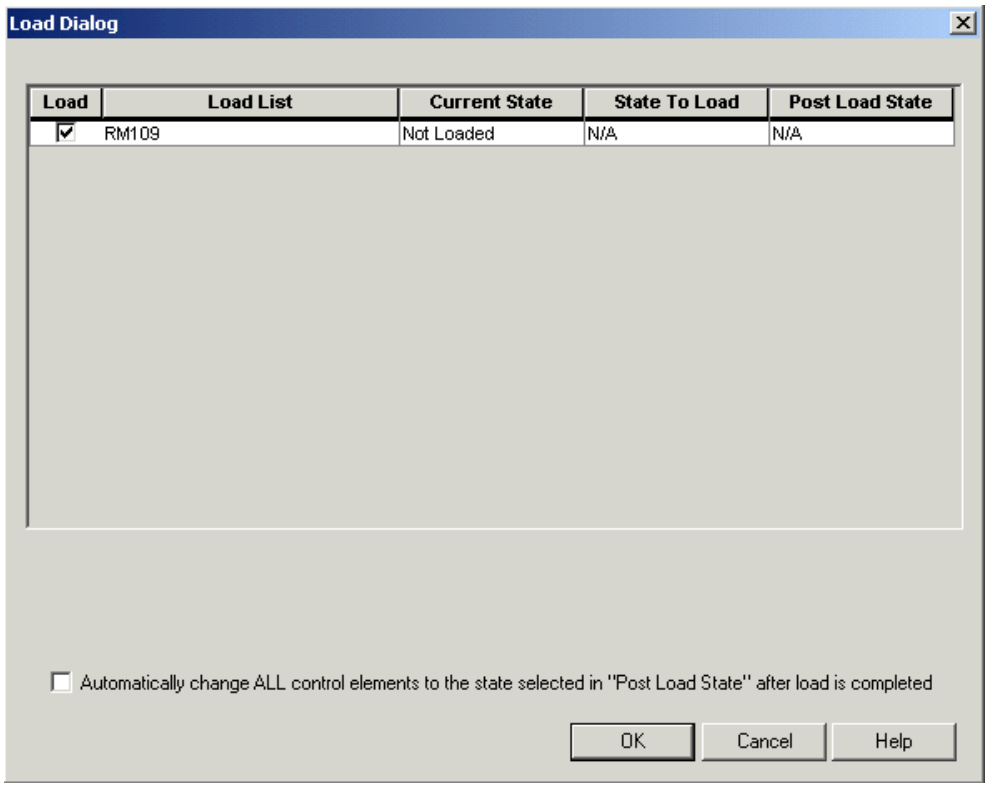
- Control Builder is running
- This procedure assumes that the RM and matching pair are installed and capable of communicating with the Server.

Considerations:

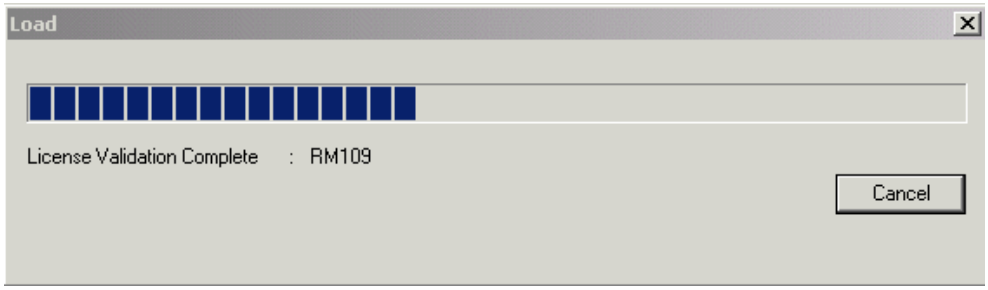
- All illustrations used in the procedure are for example purposes only.

Step	Action	Result
1	Click desired primary RM block icon in Project tab.	Selects and highlights the component. 
2	Click Tools->Load. Or, click the  load button in the toolbar.	Calls up Load Dialog box

Step	Action	Result



3	With load check box checked, click the OK button.	Initiates the load to the primary RM and calls up the load progress dialog.



Step	Action	Result
	**TIP** If errors are detected, they will be displayed in the Load progress dialog and you will be asked if you want to continue the load or cancel, depending on the nature of the error. We suggest that you cancel the load and identify and fix the errors. Each message includes an error code in parentheses. Note the last number in the string. In some cases, more information about the code number may be included in the Control Builder Notifications Reference document in Knowledge Builder.	
4	Once the load completes and the dialog box closes, click the Monitoring tab.	Primary RM icon now appears in Monitoring tab. The default state for a loaded RM is inactive or color code blue.
5	Repeat Steps 1 to 4 for the secondary RM. Select the secondary RM icon in place of the primary one.	Both primary and secondary RM icons appear in the Monitoring tab.

Loading IOP

Use the following general procedure to load an Input/Output Processor (IOP). The load procedure is similar for all I/O signal-handling components.

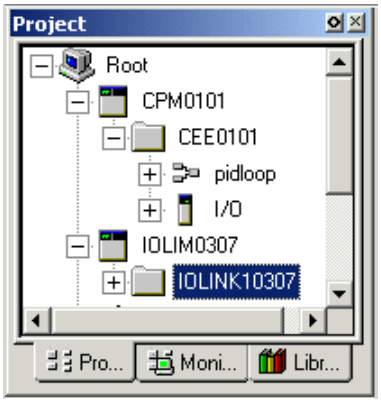
Prerequisites:

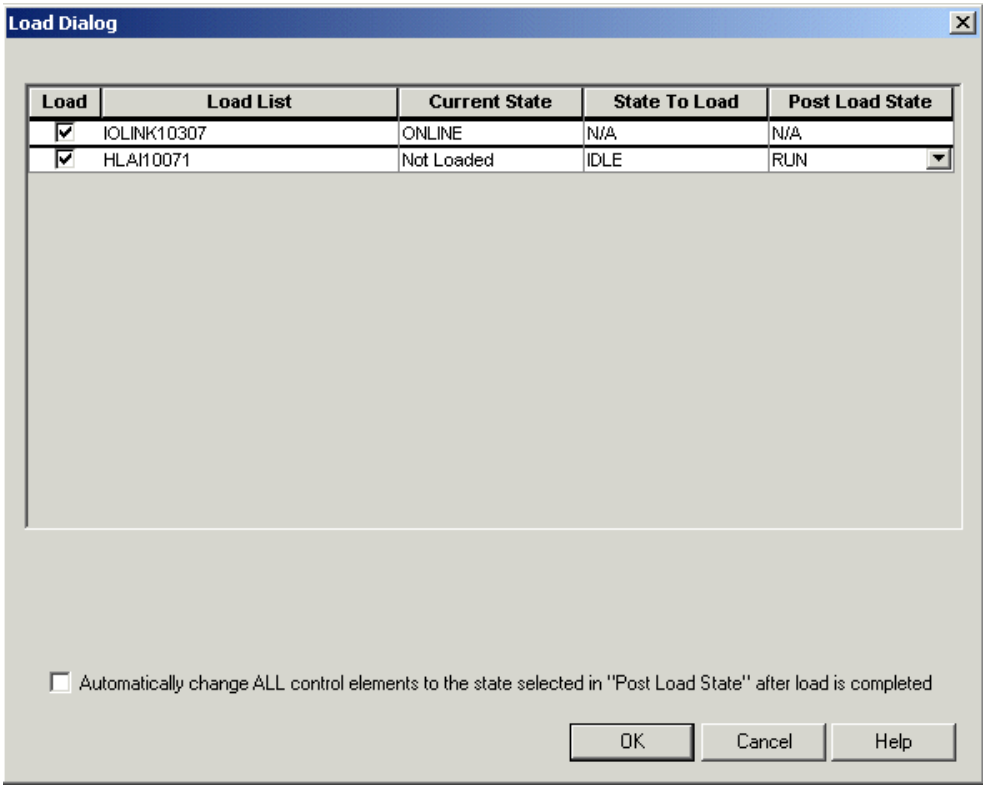
- Control Builder is running
- This procedure assumes that the IOLINK and IOP are installed and capable of communicating with the Server.

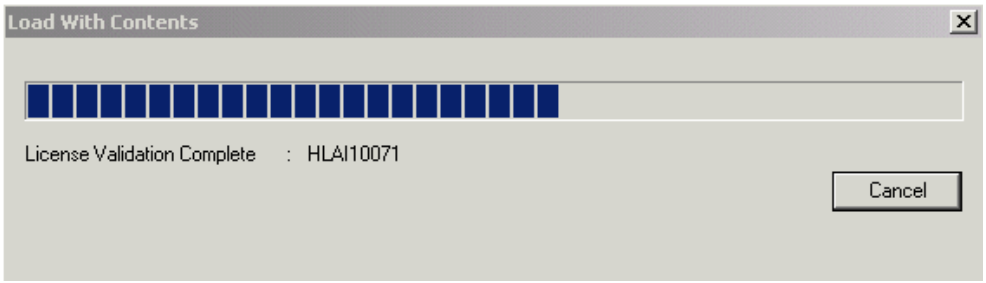
Considerations:

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

Step	Action	Result
1	Click desired IOLINK block icon in Project tab.	Selects and highlights the component.

Step	Action	Result
		
2	Click Tools->Load with Contents	Calls up Load Dialog box

Step	Action	Result
		
3	Be sure load check boxes are checked for components to be loaded. Click component to remove/insert check mark. Click the OK button.	Initiates the sequential load to the selected components and calls up the load progress dialog.

Step	Action	Result
		
	TIP If errors are detected, they will be displayed in the Load progress dialog and you will be asked if you want to continue the load or cancel, depending on the nature of the error. We suggest that you cancel the load and identify and fix the errors - unless you know the error is due to missing hardware that is scheduled to be installed later. Each message includes an error code in parentheses. Note the last number in the string. In some cases, more information about the code number may be included in the Control Builder Notifications Reference document in Knowledge Builder.	
4	Once the load completes and the dialog box closes, click the Monitoring tab.	IOP icon now appears in Monitoring tab. The default state for a loaded IOP is active or color code green.
5	Repeat this procedure to install other IOPs.	

Loading IOM

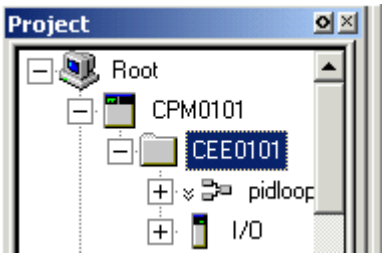
Use the following general procedure to load an Input/Output Module (IOM). The load procedure is similar for all I/O signal-handling components.

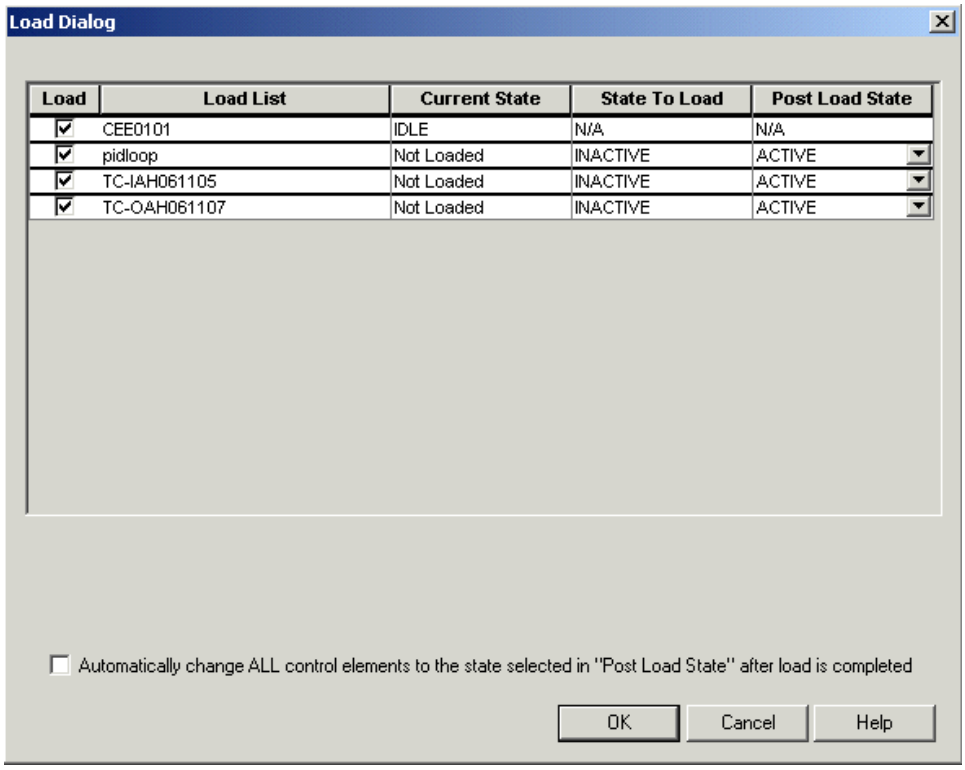
Prerequisites:

- Control Builder is running
- This procedure assumes that I/O Modules are installed and capable of communicating with the Server.

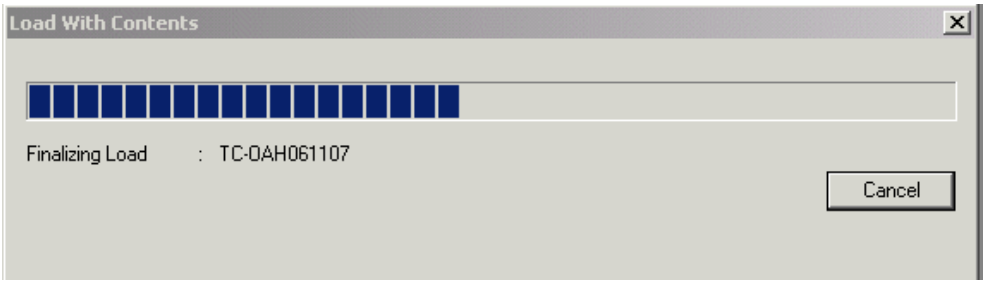
Considerations:

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

Step	Action	Result
1	Click desired CEE block icon in Project tab.	Selects and highlights the component. 
2	Click Tools->Load with Contents	Calls up Load Dialog box



3	• Be sure load check boxes are checked for components to be loaded. Click a	Initiates the sequential load to the selected components and calls up the

Step	Action	Result
	component to remove/insert a check mark. • Be sure no Control Module blocks are Checked. Click the OK button.	load progress dialog.
		
	TIP If errors are detected, they will be displayed in the Load progress dialog and you will be asked if you want to continue the load or cancel, depending on the nature of the error. We suggest that you cancel the load and identify and fix the errors. Each message includes an error code in parentheses. Note the last number in the string. In some cases, more information about the code number may be included in the Control Builder Notifications Reference document in Knowledge Builder.	
4	Once the load completes and the dialog box closes, click the Monitoring tab.	IOM icon now appears in Monitoring tab. The default state for a loaded IOM is inactive or color code blue.
5	See the Activating the CEE section in this document to set the IOM to its Run state.	IOM icon turns green when active.
6	Repeat this procedure to install other IOMs.	

Loading CM

Use the following general procedure to load a Control Module or Sequential Control Module. The load procedure is similar for all container type components.

Prerequisites:

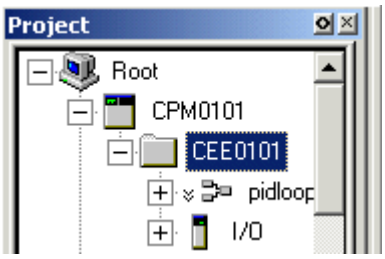
- Control Builder is running
- This procedure assumes that the CPM is installed and capable of communicating with the Server.

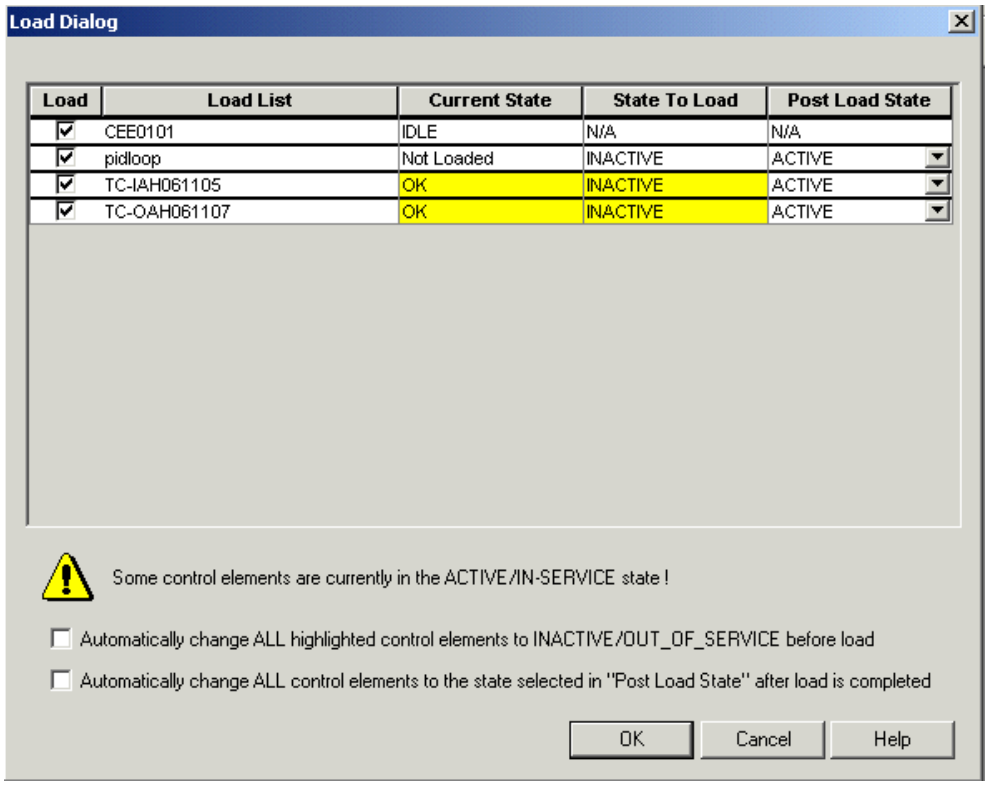
Considerations:

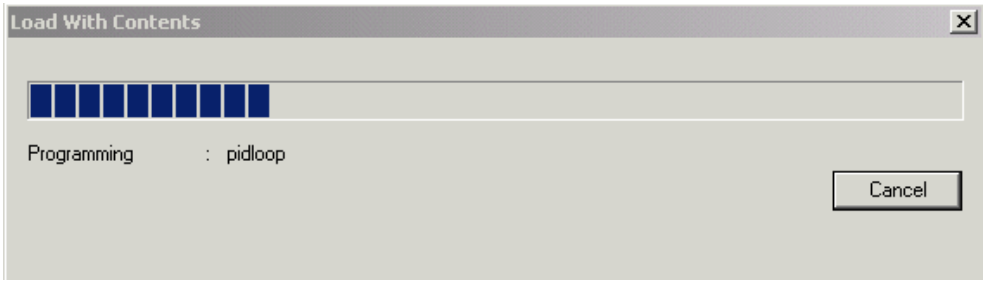
- All illustrations used in the procedure are for example purposes only.

ATTENTION

Before attempting to load any CM or SCM components, be sure its control chart is **not** open in Control Builder.

Step	Action	Result
1	Click desired CEE block icon in Project tab.	Selects and highlights the component. 
2	Click Tools->Load with Contents	Calls up Load Dialog box

Step	Action	Result
		
3	• Be sure only load check boxes for CMs or SCMs are checked for components to be loaded. Click a component to remove/insert a check mark. • Be sure to click appropriate automatic change selections, if loads are made to active installed components. Click the OK button.	Initiates the sequential load to the selected components and calls up the load progress dialog.

Step	Action	Result
		
	TIP If errors are detected, they will be displayed in the Load progress dialog and you will be asked if you want to continue the load or cancel, depending on the nature of the error. We suggest that you cancel the load and identify and fix the errors. Each message includes an error code in parentheses. Note the last number in the string. In some cases, more information about the code number may be included in the Control Builder Notifications Reference document in Knowledge Builder.	
4	Once the load completes and the dialog box closes, click the Monitoring tab.	CM icon now appears in Monitoring tab. The default state for a loaded CM is inactive or color code blue.
5	See the Activating the CEE section in this document to activate the CM.	CM icon turns green when active.
6	Repeat this procedure to install other CMs or SCMs.	

Re-loading components

You can use the applicable previous load procedure to re-load data to components from the Project or Monitor tab. It is good idea to invoke the following commands through the TOOLs menu after a re-load operation.

- Update to Project
- Upload

Using Upload command

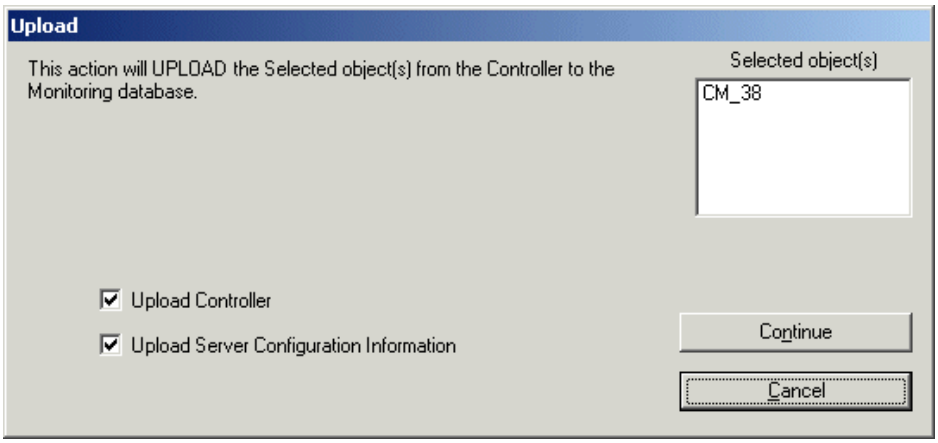
Prerequisites:

- Control Builder is running
- Tree windows are open

Considerations:

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

Use the following procedure as a guide to initiate an **Upload** command.

Step	Action
1	In **Monitoring** tree, click icon for tagged object such as Control Module (CM) or Sequential Control Module (SCM)
2	In **Tools** menu, click **Upload**. Or, right-click the icon and select **Upload** from the shortcut menu.
	
3	In **Upload** dialog, • be sure the **Upload Controller** check box is checked (default selection), if you want to upload data for the selected object from the Controller to the Monitoring Engineering Repository Database (ERDB). Clear the check box to disable the upload of data from the Controller. • be sure the **Upload Server Configuration Information** check box is

Step	Action
	checked (default selection), if you want to upload data for the selected object from the Experion Server to the ERDB. Clear the check box to disable the upload of data from the Server.
4	In **Upload** dialog, click the **Continue** button to initiate the upload.
	**ATTENTION** • When **Upload Server Configuration information** check box is checked, configuration data uploaded from the Experion Server database overwrites any existing configuration data in the Monitoring ERDB. This includes information configured through the **Server History** and **Server Display** configuration forms for the tagged object in the Project tree. • To transfer configuration data uploaded to the Monitoring ERDB to the Project ERDB, click **Tools**->**Update to Project** with object icon selected in **Monitoring** tree.
5	Repeat these Steps to upload data for other tagged objects.

Using Upload With Contents command

Prerequisites:

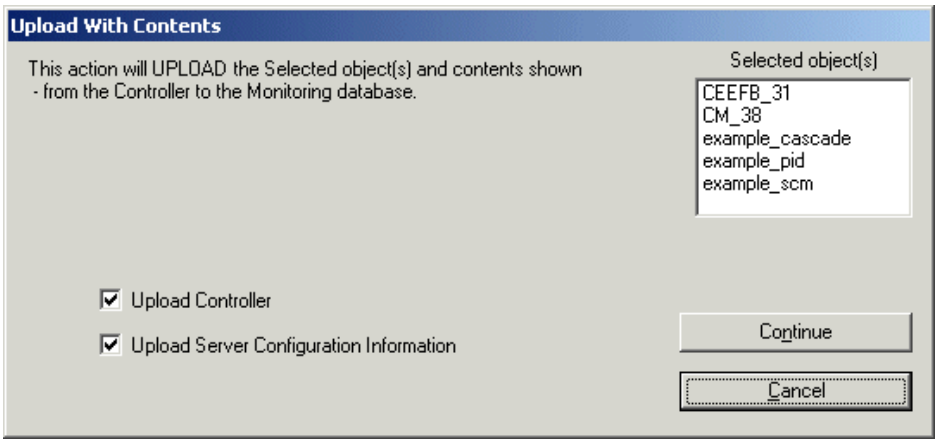
- Control Builder is running
- Tree windows are open

Considerations:

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

Use the following procedure as a guide to initiate an **Upload With Contents** command.

Step	Action
1	In **Monitoring** tree, click icon for Control Execution Envirnoment function block, such as CEEFBxx.
2	In **Tools** menu, click **Upload With Contents**. Or, right-click the icon and select **Upload With Contents** from the shortcut menu.

Step	Action
	
3	In **Upload With Contents** dialog, • be sure the **Upload Controller** check box is checked (default selection), if you want to upload data for the selected object(s) from the Controller to the Monitoring Engineering Repository Database (ERDB). Clear the check box to disable the upload of data from the Controller. • be sure the **Upload Server Configuration Information** check box is checked (default selection), if you want to upload data for the selected object(s) from the Experion Server to the ERDB. Clear the check box to disable the upload of data from the Server.
4	In **Upload With Contents** dialog, click the **Continue** button to initiate the upload.
	ATTENTION • When **Upload Server Configuration information** check box is checked, configuration data uploaded from the Experion Server database overwrites any existing configuration data in the Monitoring ERDB. This includes information configured through the **Server History** and **Server Display** configuration forms for the tagged object(s) in the **Project** tree. • To transfer configuration data uploaded to the Monitoring ERDB to the Project ERDB, click **Tools->Update to Project** with object icon selected in **Monitoring** tree.
5	Repeat these Steps to upload data for other Control Execution Envirnoment

Step	Action
	blocks.

Using Compare Parameters

Purpose of Compare Parameters

Compare Parameters extends the load functionality of Control Builder to allow the user to decide which parameters will and will not be downloaded based on the following criteria:

- The parameter must be controller (CEE) resident, or system repository (SR) resident.
- The parameter being downloaded (from the database) must be different from the value in the controller prior to the download

After the user selects one or more objects to download they will be presented with a dialog box that allows them to:

- Look at a list of parameters that meet the above criteria and decide which parameters they wish to download
- Print out a report showing the list of differences (user may continue with load after the report is printed)
- Get help regarding the new functionality

Without having to perform a load function, the user may:

- Select and view the differences for several objects (same criteria as above)
- Print a report showing these differences

Enabling Compare Parameters

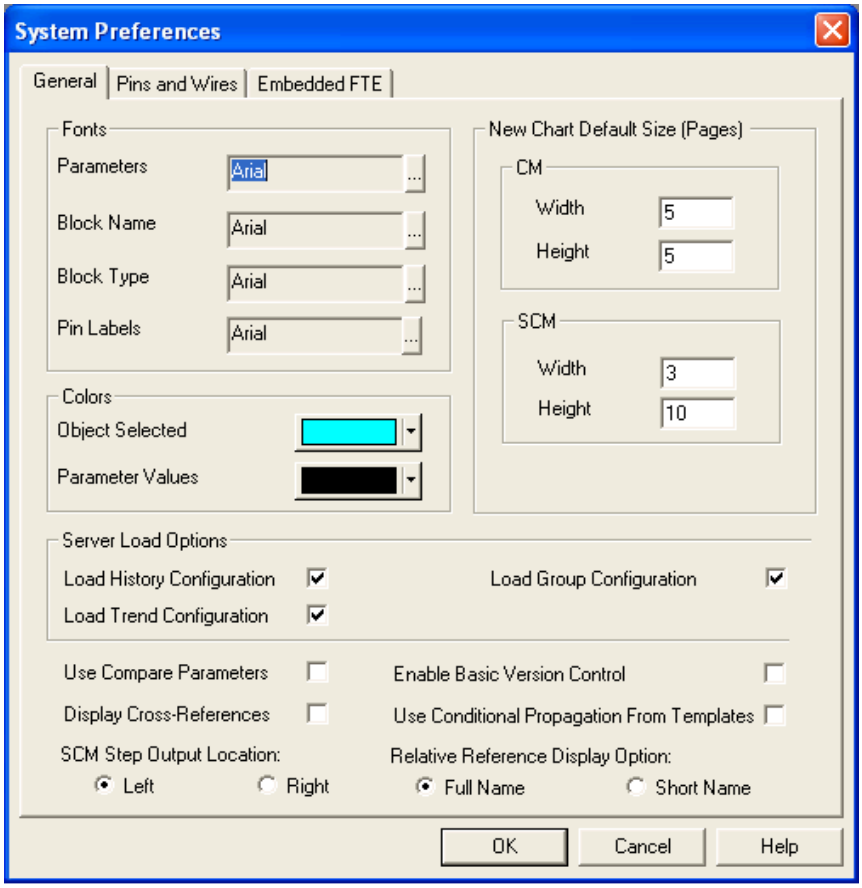
Compare Parameters functionality is available immediately after an Experion system is installed (which includes engineering tools) and requires no special licensing.

Prerequisites:

- Control Builder is running
- Tree windows are open

Considerations:

- A "Use Compare Parameters" check box available in the System Preferences dialog allows Compare Parameters to be turned Enabled or Disabled. By default, Compare Parameters will be disabled upon installation.
- All illustrations used in the procedure are for example purposes only.

Step	Action
1	Start Control Builder. Click Tools-> System Preferences to open the dialog.
	
2	On the General tab, verify the Use Compare Parameters check box is **checked** or enabled. Click the check box, if it is **blank** or disabled.

Step	Action
3	Click OK to close the System Preferences dialog.

Compare parameters security

For the use of Compare Parameters within a load, the access level will be the same for Compare Parameters as it is to perform a load

For the use of Compare Parameters outside of a load (standalone) the access level will be Operator.

Using Compare Parameters during a load

Prerequisites:

- Control Builder is running
- Tree windows are open

Step	Action
1	With Compare Parameters enabled, select modules from the tree in Control Builder: • From the Project or Monitor trees • From assignment view or hierarchical view
2	Issue a load command by: • Right-clicking on the selected modules in the tree • Selecting the load icon from the toolbar • Selecting load from the main menu
3	The LoadUI dialog box appears. Select which modules will be loaded and what the post load states will be. Close the dialog box by selecting the OK button.
4	The Compare Parameters Dialog box will open. Select which parameters will be downloaded to the controller and which will not. Click the OK button. Refer to "*Compare Parameters Dialog box*" for additional information.

Step	Action
5	The load will process. When complete, review entries in the Experion event log.

Using Compare Parameters during a load



TIP

When the user does not load a parameter to the controller this parameter is not uploaded to the project or monitoring database.

Standalone functionality for Compare Parameters is the ability to generate a list of differences (Controller vs. database) for objects that is identical to those seen during the load operation, without performing a load.

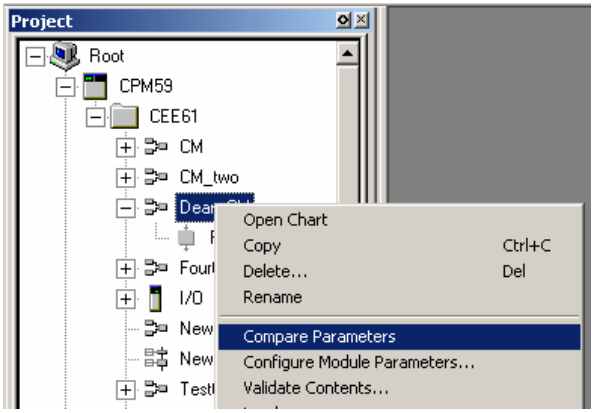
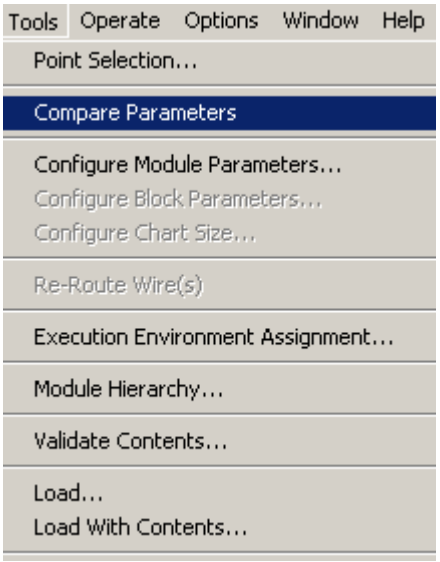
Prerequisites:

- Control Builder is running
- Tree windows are open

Considerations:

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

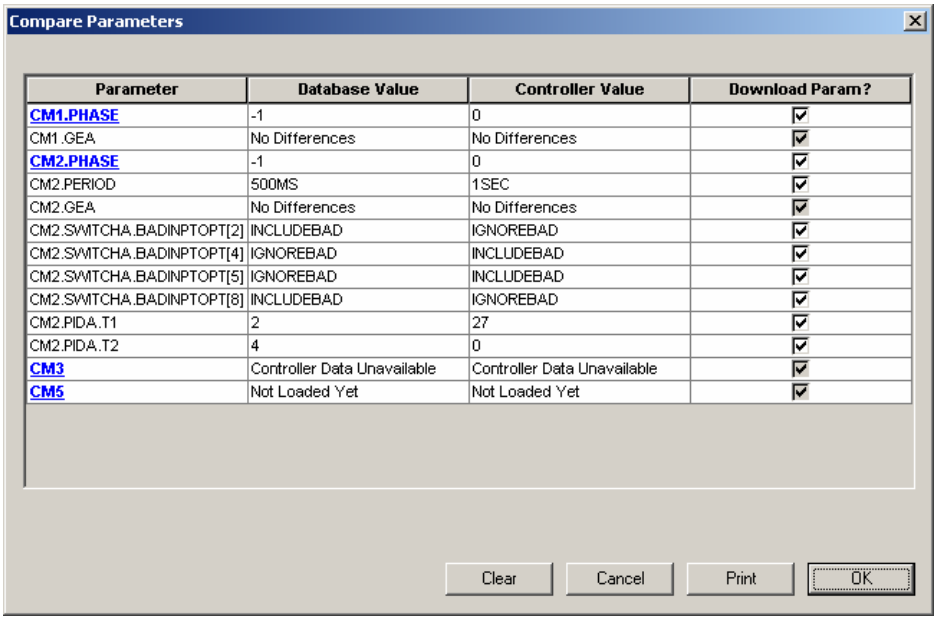
Step	Action
1	With Compare Parameters enabled, select modules from the tree in Control Builder: • From the Project or Monitor trees • From assignment view or hierarchical view
2	Right-click on the module and select ‘Compare Parameters’…

Step	Action
	 Or select 'Compare Parameters from the Tools menu. 
3	Once the objects are selected and the Compare Parameters is initiated, the Compare Parameters dialog box will appear. Refer to "*Compare Parameters Dialog box*" for additional information.

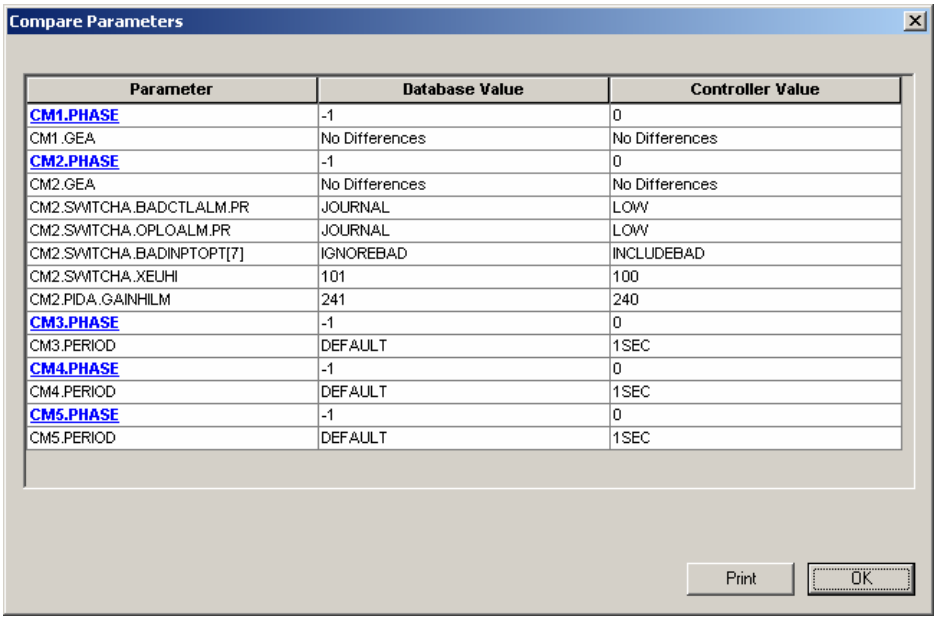
Step	Action
	Print or close the dialog box when finished.

Using the Compare Parameters dialog box

The Compare Parameters Dialog box will vary slightly depending on the situation. When using Compare Parameters when not performing a Load, the Clear and Cancel buttons, and 'Download Param?' column are not available.



Compare Parameters dialog box during load operations



Compare Parameters Dialog box when a load is not being performed.

Dialog box functions

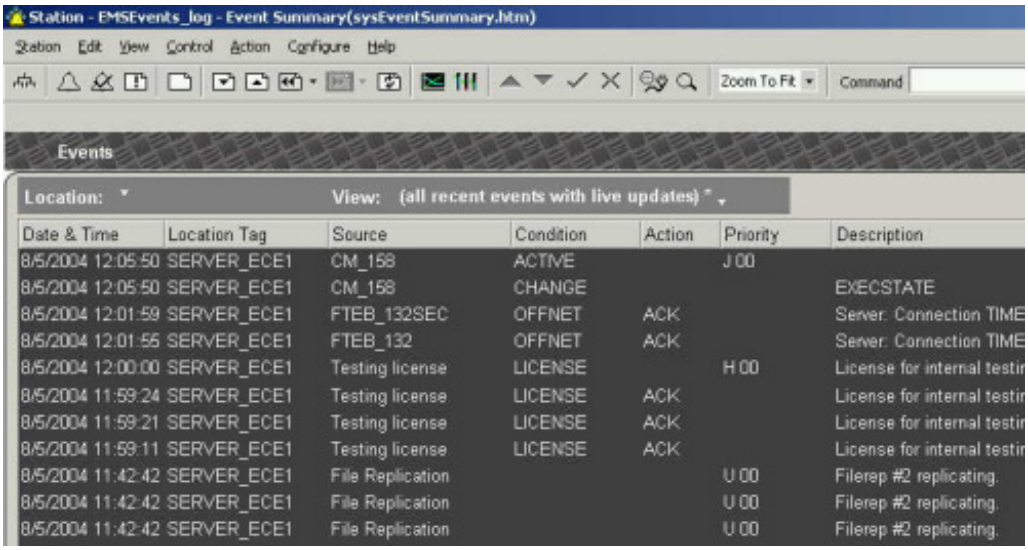
Column name	Function
Parameter	Shows parameters from modules (tagged blocks) where the database value is different than the controller value. The parameter name will be shown in full form containing both the module name as well has the parameter and intermediate parameter names.
Database Value	The value of the parameter from the database.

Column name	Function
Controller Value	The value of the parameter from the controller prior to completion of loading with the following exceptions: • If an object being loaded has no differences between the database and the controller value then the words 'No differences found' will appear in both the Database and Controller Value columns. • If an object being loaded is a newly created object, which has never been loaded before, the words 'Not Loaded Yet' will appear in both the Database and Controller Value columns. • If an object has been loaded before, but there is another object within it that has been newly created and never loaded before, the words 'Not Loaded Yet' will appear in both the Database and Controller Value columns for the basic block. • If an object cannot be accessed, the words 'Controller Data Unavailable' will appear.
Download Param?	A checkbox where the user can select to download (checked) or not download (unchecked) the database value to the controller. If the checkboxes are grayed out, the user must download the parameters. This column is not available when Compare Parameters is used without a performing a Load.
Print button	Prints the currently viewed values into a report. A report generator by this method does not require object selection because object names and data are already known.
Clear button	Deselects all check boxes for all entries in the table. Not available when Compare Parameters is used without performing a Load.
Cancel button	End (aborts) the load process. Prompts if modules should or should not be set to their post load state. Not available when Compare Parameters is used without performing a Load.

Column name	Function
OK button	Signals that all selections on the form have been made and that the load should continue.

Event log entries during a controller load

When a controller load is performed and not all parameters are selected for download, entries are made in the Experion event log. The entries indicate which parameters the user decided not to download to the controller.



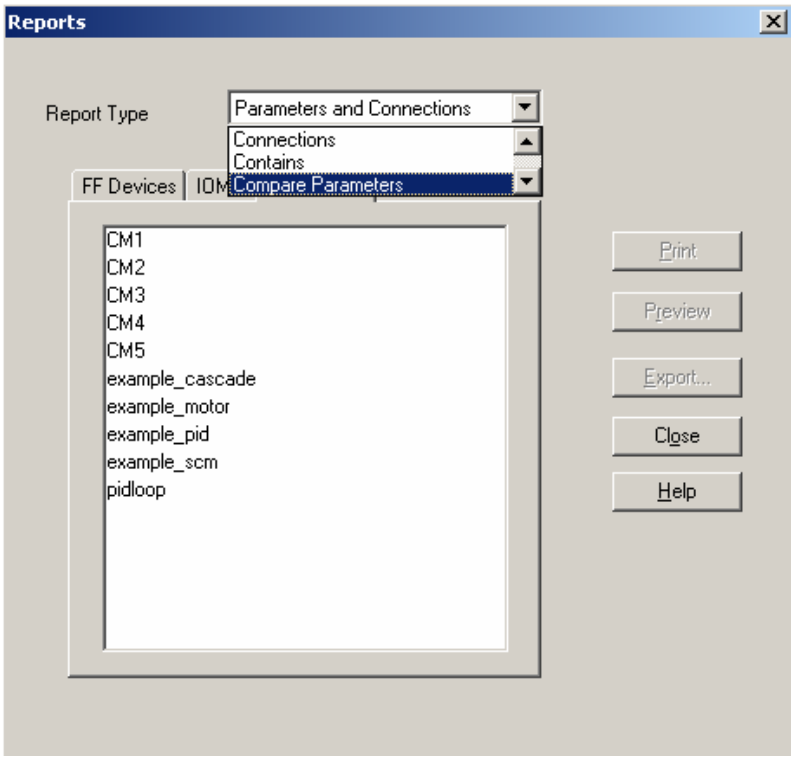
Event log columns

Column name	Functionality
Date & Time	Date and time when the parameter was bypassed from loading.
Location Tag	Name of location tag
Source	Name of the source
Condition	Current condition
Action	The current action

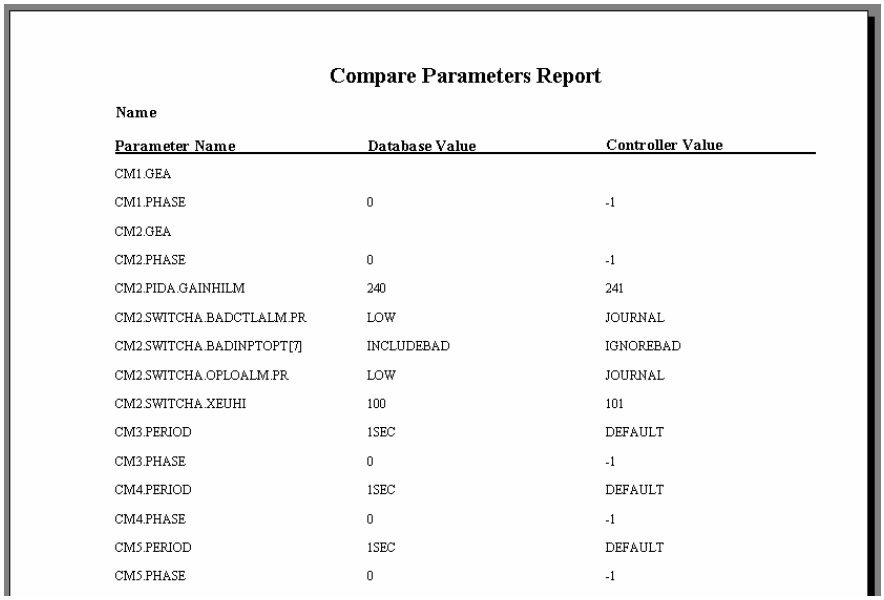
Column name	Functionality
Priority	The priority setting
Description	Description of the event's stats
Additional columns are displayed	

Using Compare Parameters reports

Compare Parameters reporting is available as a selection in Control Builder report generation.

Step	Action
1	Call up the Report Selection box.
2	Select Compare Parameter as the Report Type. 

Step	Action
3	Select which object to report on from the tabs available.
4	Click Print to create the report.



Compare Parameters Report

Name

Parameter Name	Database Value	Controller Value
CM1.GEA		
CM1.PHASE	0	-1
CM2.GEA		
CM2.PHASE	0	-1
CM2.PIDA.GAINHILM	240	241
CM2.SWITCHA.BADCTLALM.PR	LOW	JOURNAL
CM2.SWITCHA.BADINPTOPT[7]	INCLUDEBAD	IGNOREBAD
CM2.SWITCHA.OPLOALM.PR	LOW	JOURNAL
CM2.SWITCHA.XEUHI	100	101
CM3.PERIOD	1SEC	DEFAULT
CM3.PHASE	0	-1
CM4.PERIOD	1SEC	DEFAULT
CM4.PHASE	0	-1
CM5.PERIOD	1SEC	DEFAULT
CM5.PHASE	0	-1

Compare Parameters Report

Compare Parameters Report columns

Column name	Functionality
Parameter Name	Shows parameters from modules (tagged blocks) where the database value is different than the controller value. The parameter name will be shown in full form containing the module, parameter, and intermidiate parameter names.
Parameter Database Value	The value of the parameter in the database (Monitor or Project view).

Column name	Functionality
Controller Value	The value of the parameter in the controller (snapshot).

Copying control strategies using templates

Background

Templates may be created in Control Builder. Once created, templates may be used to create multiple copies of desired control strategies.

A template is a user-configured control strategy (that is, a logically specified set of function blocks arranged on a Control Builder control drawing to simulate your process application) that does not have I/O assignments, parameter connections or SCM expressions explicitly declared outside the scope of the copy operation.

- Inside the scope of operation means that the specific control object (such as a function block, I/O assignment, parameter connection or SCM expression) is included in the list of selected control objects to be copied.
- Outside the scope of operation means that the specific control object is not included in the list of selected control objects to be copied.

Notes:

- The user-configured template uses parameters with substitute names in its expressions.
- After copying a desired control strategy using a template, you must replace (resolve) each substitute name with an appropriate block name for that particular control strategy as described in the subsequent procedure titled Using a Template.
- Substitute names must be resolved prior to loading the control strategy to the controller.

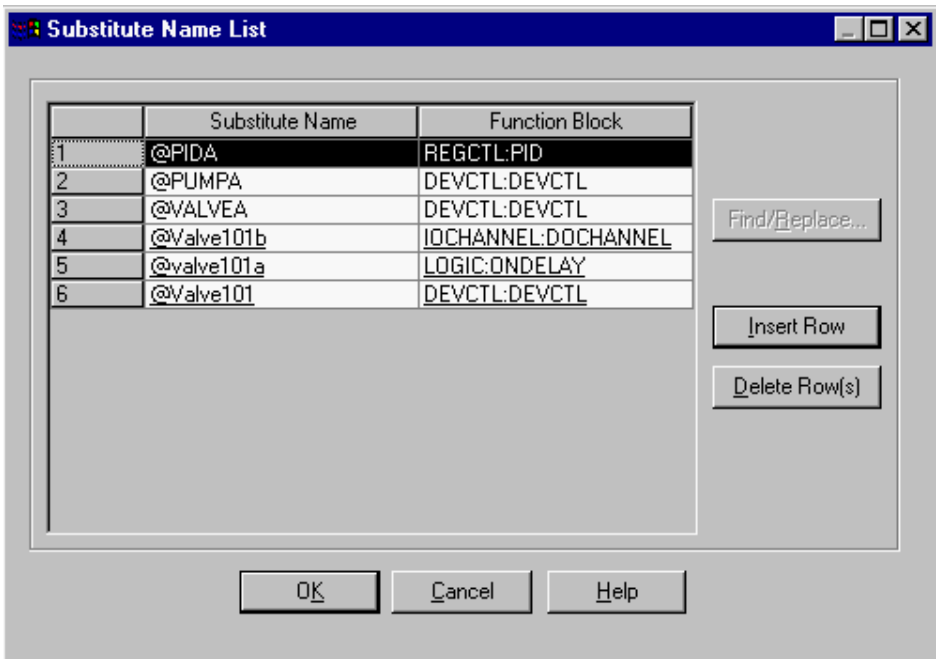
Creating a substitute name list

Follow the steps in the table below to create a substitute name list to be used in a user-configured template.

ATTENTION

- Substitute names must be created prior to being used in a control strategy.
- Substitute names must have the following format: @xxxxx….
- Every substitute name must begin with the "@" character.
- Substitute names may be up to 64 characters in length, including the "@" sign and at most one (1) period.
- A substitute name represents the first two parts of a parameter tagname or a basic block: **[CM.BLOCK]**.PARAMETER where **[CM.BLOCK]** represents the substitute name.

Step	Action	Result
1	Select Tools -> Substitute Name List to call up the Substitute Name List dialog. Alternate method: Select the  toolbar button.	The Substitute Name List dialog appears (see the following Figure).
2	Create a substitute name for each function block to be referenced by the generic control strategy. • Click Insert Row to add a new row for each substitute name to be added.	The Substitute Name List dialog becomes populated with the substitute names required for the template to be created.

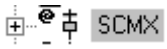


Substitute name list dialog

Creating a template

Follow the steps in the table below to create a template to be used to create a desired number of copied control strategies.

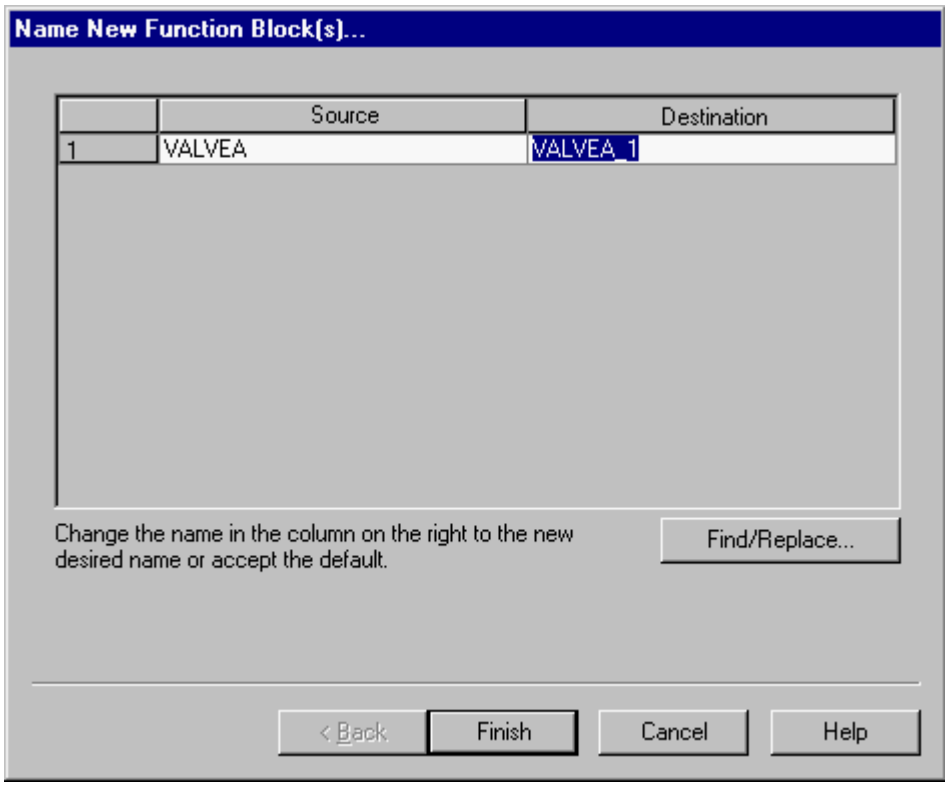
Step	Action	Result
1	Create a generic control strategy using parameter connectors or SCM expressions with substitute names.	The template is created and resides on the Project tree. For example, a control strategy template named “SCMX” might be created and located on the Project Tree. • An “@” character is attached to the “SCMX” name on the Project Tree to indicate that some of its expression components use substitute names which will be

Step	Action	Result
		resolved during a copy procedure using the Operation Name List Wizard. 

Using a template

Follow the steps in the table below to use a previously-created template to create a desired number of copied control strategies.

Step	Action	Result
1	Select the template to be copied on the Project tree. For example, a control strategy template named “SCMX” might be located on the Project tree. This template was created previously and has a “@” character appearing next to it on the Project Tree.	The template is highlighted on the Project tree.
2	Select Edit -> Copy. Alternate methods: • Click <Ctrl>-C • Select the Copy toolbar button  .	The template is copied to the Control Builder clipboard. The Name New Function Block(s) dialog appears (see foillowing figure). Note: This is the first page of the Operation Name List Wizard.



Name new function block(s) dialog

ATTENTION

The following connection rules apply when copying function blocks:

Notes:

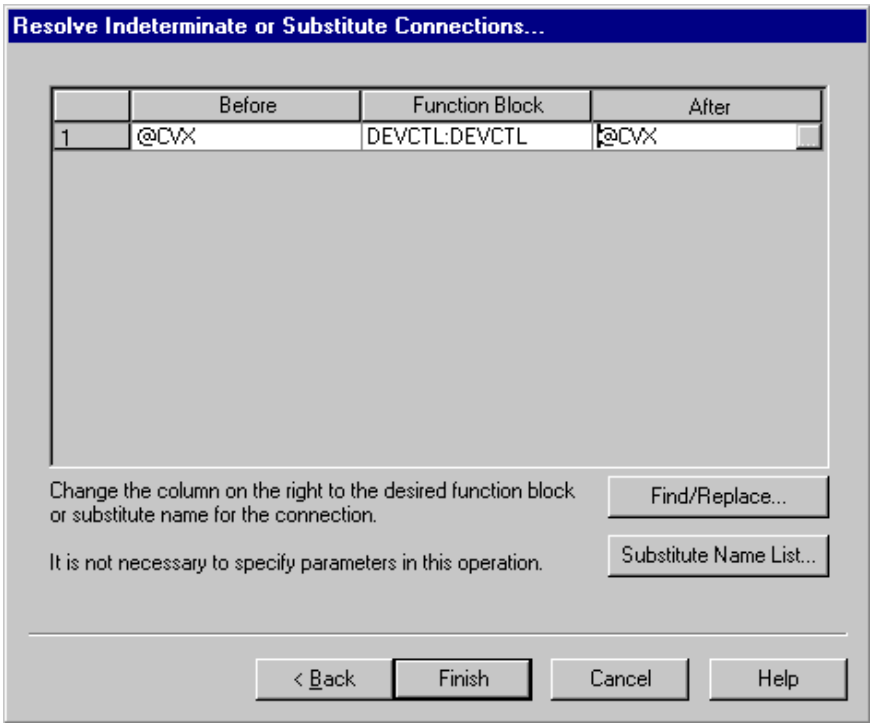
Inside the scope of operation means that the block is included in the list of selected blocks to be copied.

Outside the scope of operation means that the block is not included in the list of selected blocks to be copied.

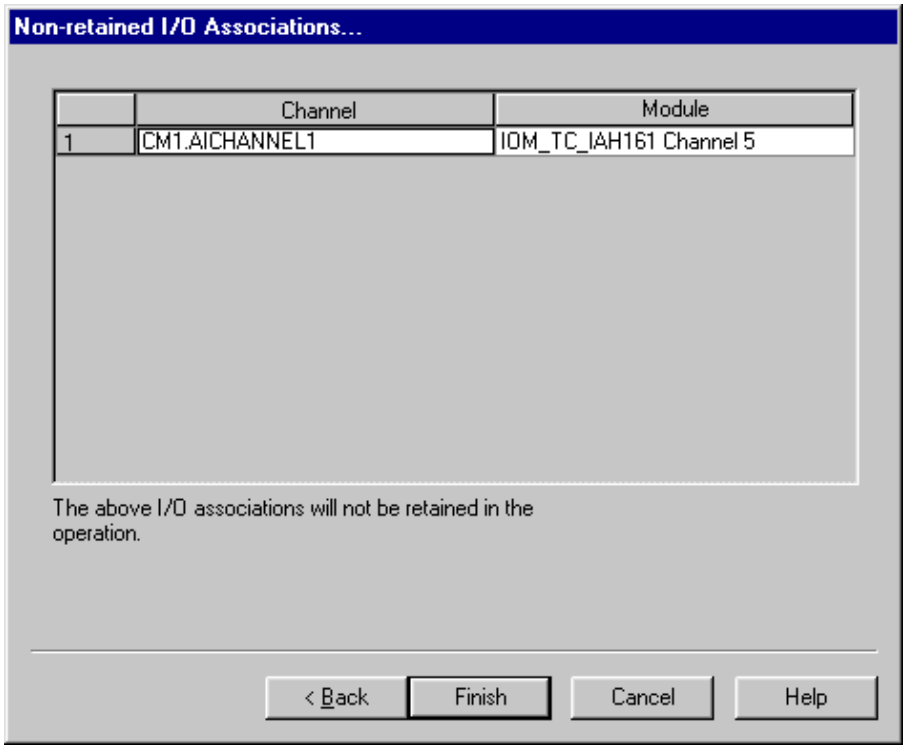
1. Graphical connections to blocks that are inside the scope of operation are automatically resolved relative to the operation.
2. Graphical connections to blocks that are outside the scope of the operation are not carried over to the new copied blocks.
3. Parameter connections to blocks that are inside the scope of operation are automatically resolved relative to the operation.
4. Parameter connections to blocks that are outside the scope of operation appear in the Connections page to be resolved by the user if desired at the time of copy. An exception would be parameter connections which have hidden “back init” which are discarded. IO assignments are also discarded and shown in a special window.
5. Substituted connections on selected blocks always appear in the Connections Page to be resolved by the user if desired at the time of copy. The same applies to references.

3	Change the control strategy’s name in the Destination column of the Name New Function Block(s) dialog (see the following Figure) to a desired name or accept the default name.	The copied control strategy is assigned a new name.

4	Click Next or Finish as appropriate.	If appropriate, additional pages of the Operation Name List Wizard appear, depending on the particular control strategy being copied, to include: • Resolve Indeterminate or Substitute Connections dialog (see the following Figure). • Resolve Deleted I/O Associations (see the following Figure).



Resolve indeterminate or substitute connections dialog



Non-retained I/O associations dialog

5	Follow the directions on each additional page of the Operation Name List Wizard. Click Finish on the final page of the wizard to complete each copy operation.	The copied control strategy is pasted to the Project Tree.
6	Repeat steps 1-5 to make additional copies of control strategies as required.	Additional control strategies are created.

Using the Change Parent Template wizard

Change Parent Template functional description

The goal of the Change Parent Template operation is to provide the capability to change the derivation hierarchy of a control module, embedded block, or user template that had been derived from a user template.

The selected new parent must have been included in the original derivation hierarchy of the target object. For example, if strategy “X” is derived from user template “Kid”, and “Kid” is derived from user template “Dad”, and “Dad” is derived from user template “Grandpa”, and “Grandpa” is derived from the System “Control Module” template, then, the valid new parent choices for kid “X” would be either “Grandpa” or the System “Control Module”.

The Change Parent Template wizard is invoked to change the parent template of:

- a user template,
- a stand-alone template embedded in a user template, and which has not been inherited from the parent of the user template,
- a control module derived from a user template, and
- an embedded block contained in a strategy that was derived from a user defined stand-alone template.

Change Parent Template requirements

The ENGR level of access and the User Templates Licensed Option are required to use the Change Parent Template Wizard.

Invoking the Change Parent Template wizard



WARNING

Do not change the content or structure of the ERDB with respect to user-defined templates while the Change Parent Template Wizard is in operation.

The system does not prevent the ERDB from being modified, or blocks from being locked that are needed by an open Change Parent Template Wizard".

If the problem is one of database structure change while the wizard is in progress, it is possible for the wizard to present information, including selection possibilities for objects that have since been removed from the ERDB. If the wizard property page does not have valid selections, immediately exit the wizard, and DO NOT select objects that are no longer contained in the ERDB.

There are two methods provided to invoke the Change Parent Template Wizard

a) Select the block for which a Change Parent Template operation is desired; or

b) Select the parent of the block for which a Change Parent Template operation is desired.

Using method (1) is recommended if there may be other instances or sub-templates of the block's parent that should also be changed.

For a Control Module Instance or for an Embedded Block in a Control Module Instance that has been derived from a Stand Alone Template:

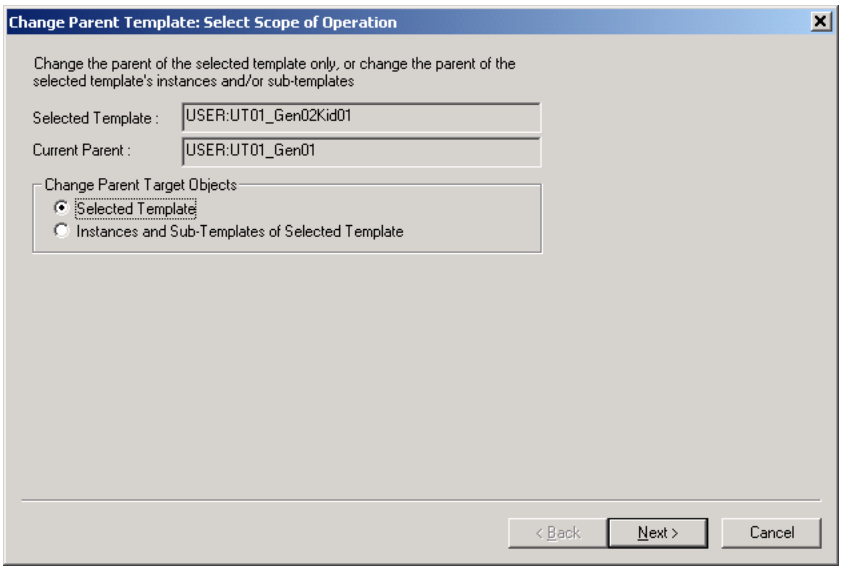
Step	Action
1	Select the Control Module on the Project tree view.
2	Templates-> Change Parent Template or right-click and select Change Parent Template.
3	The New Parent Selection page is opened.

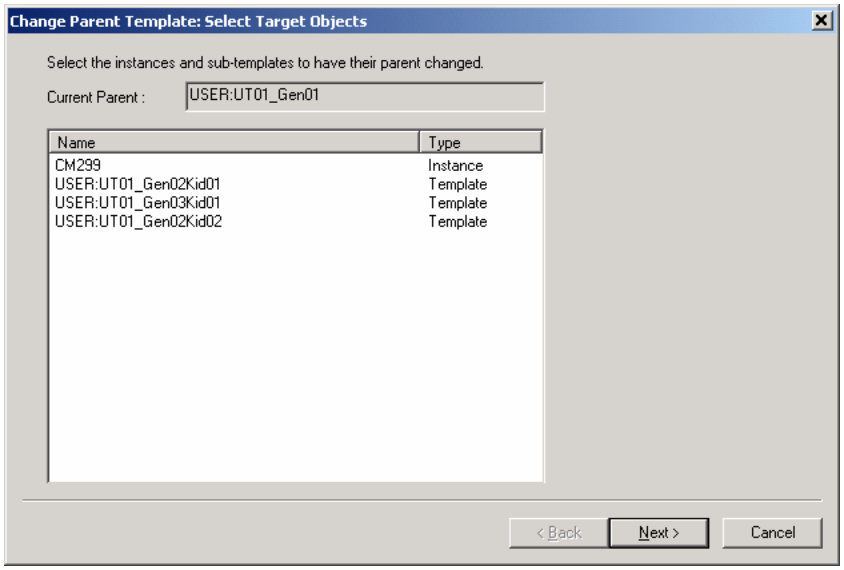
Notes:

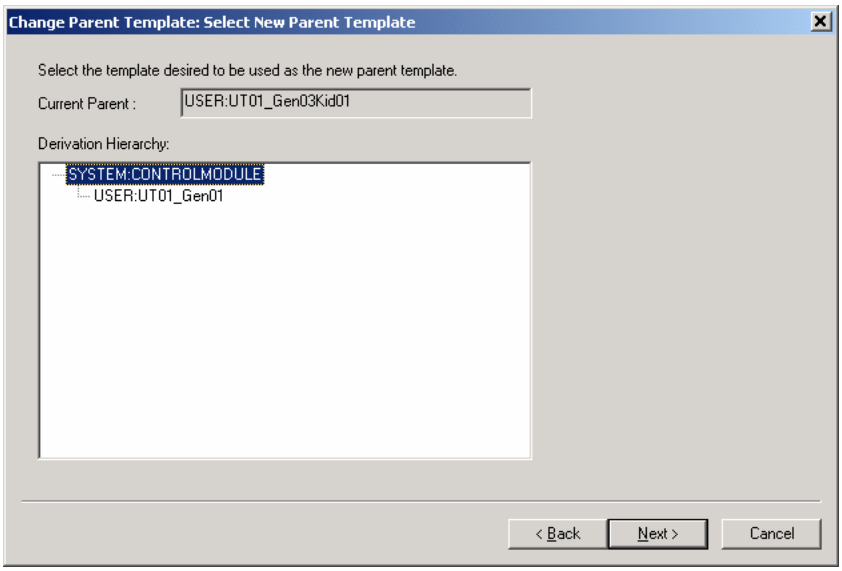
- The menu item will be disabled if more than one item is selected on the project tree.

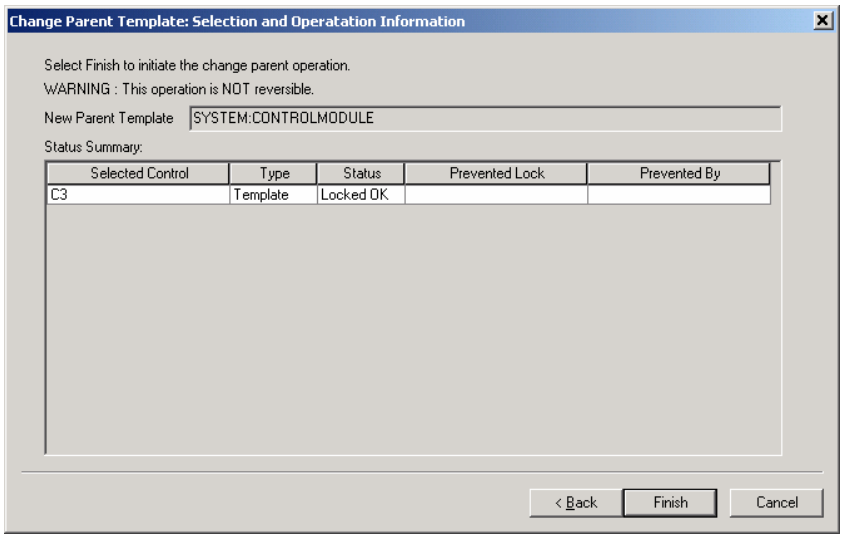
- The menu item will be disabled if the selected control module is directly derived from a System Template.
- The menu item will be disabled if the user does not have sufficient access rights to perform the Change Template Parent operation.

For a User Template, Embedded Stand Alone Template, or Sub-Template; or for an Embedded Block in a User Template that has been derived from a Stand Alone Template:

Step	Action
1	Select the target user template on the Library tree.
2	Templates -> Change Parent Template or right-click and select Change Parent Template.
3	The Select Scope of Operation page will open If the target: • is derived from a user template, and • has sub-templates and/or strategies derived from it  This page presents two radio buttons for the customer to specify if it is their desire to:

Step	Action
	• change the derivation parent of the selected template itself, or • change the derivation parent of sub-templates and/or instances of the selected template. After selecting the desired radio button, select the Next button to continue with the wizard, or the Cancel button to leave the wizard. Notes: • The menu item will be disabled if more than one template is selected is selected on the tree. • The menu item will be disabled if the template selected is not a user template. • The menu item will be disabled if the user does not have sufficient access rights to perform the Change Template Parent operation.
4	The Select Target Objects page is invoked if sub-templates and/or instances need to be selected to be the change parent operation targets. Multiple selections are permitted.  After selecting the desired target, select the Next button to continue with the

Step	Action
	wizard, or the Cancel button to exit the wizard. If another Control Builder user is modifying the database at the same time, the dialog box may open with no contents. If this occurs, cancel out of the Wizard.
5	The New Parent Selection Page allows the new parent to be selected. The current parent is indicated.  After selecting the desired new parent, select the Next button to continue with the wizard, or the Cancel button to exit the wizard. If the selected new parent template is locked by another user, an error message will be given, and the wizard will not be able to progress to the next page. If that occurs, contact the other user to unlock the new parent template before continuing with the wizard. If desired, the Cancel button may be used to cancel out of this wizard.
6	The Operation Status Page indicates all choices made, and initiates the Change Parent Template operation.

Step	Action
	 • The Selected Control column indicates the target of the operation. • The Type Column indicates if the control is a template, or strategy. • The Status column indicates the status of the operation. The following values may appear in the Status column: – Locked OK - ready for the Change Parent Template operation – Open Failed - could not open the corresponding control, will FAIL the Change Parent Template operation – Lock Failed - could not lock the corresponding control, will FAIL The Change Parent Template operation. – OK To Commit - Change Parent Template operation completed, but not yet committed to the database – OK - Change Parent Template operation completed, and committed to the database – Fail - Change Parent Template operation or commit to database has failed. If the selected control could not be locked to perform the Change Parent Template operation, the Prevented Lock column indicates the object that prevented the lock from occurring. The Prevented By column indicates the

Step	Action
	user who has locked the object that is preventing the wizard from being given the lock. After reviewing the selected targets and status, select the Finish button to commence the Change Parent Template operation. (Once the Finish button is selected, you will not be allowed to cancel out of or close the wizard until the operation completes.) Alternatively, select the Cancel button to exit the wizard, or use the Back button page back through the wizard to review previous selections. When the operation is complete, the Status column is updated accordingly. Select “Done” to exit the wizard.

Loading only server configuration

Follow the steps in the table below to load Server data only.

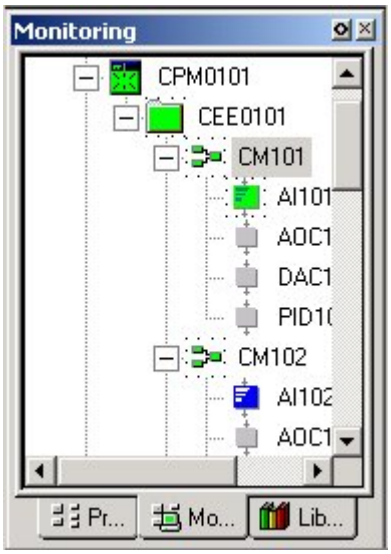


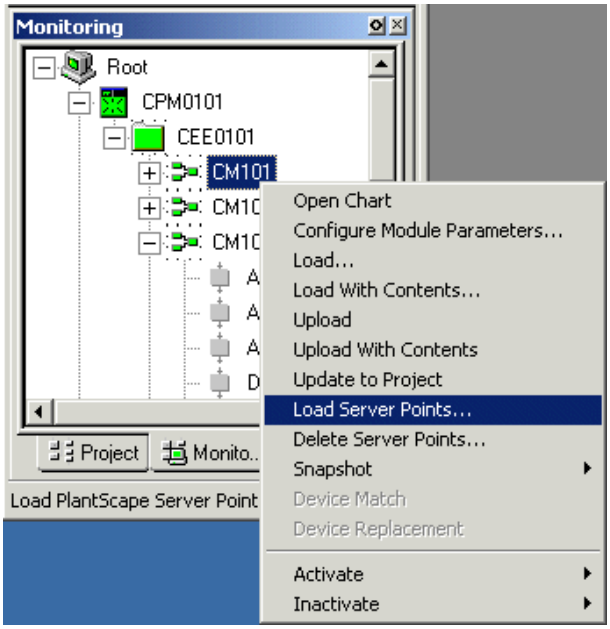
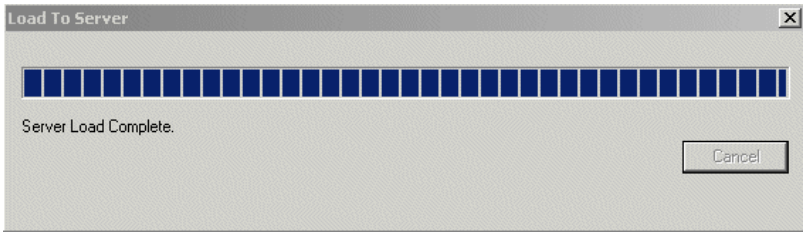
ATTENTION

Two reasons why you would want to do this is:

- If you do not have any controllers yet and you want to be able to see your tag names in selection lists in Display Builder so that you can easily build displays when the hardware is not present
- You want to reload changed server data (such as display names) without having to reload to a controller which necessitates inactivating part of your strategy

Step	Action
1	Click on the Monitoring tab; then click on the desired items in the tree to be loaded to the server. • Use either the <Shift> key or the <Ctrl> key to select multiple items in the tree to be loaded to the server. – Hold down the <Shift> key to select multiple items as an uninterrupted block in the entire list of available items. – Hold down the <Ctrl> key to select multiple items in various locations throughout the entire list of available items. Note: There is no “with Contents” for loading server points - only the

Step	Action
	selected points are loaded, not their contents as well. 

Step	Action
2	Click on Controller ->Load Server Points. 
3	Click Continue on the Load To Server dialog.
4	Control Builder will validate and rebuild the Server points based on the selected points in the Monitoring Tree. 

Using Bulk Build utility

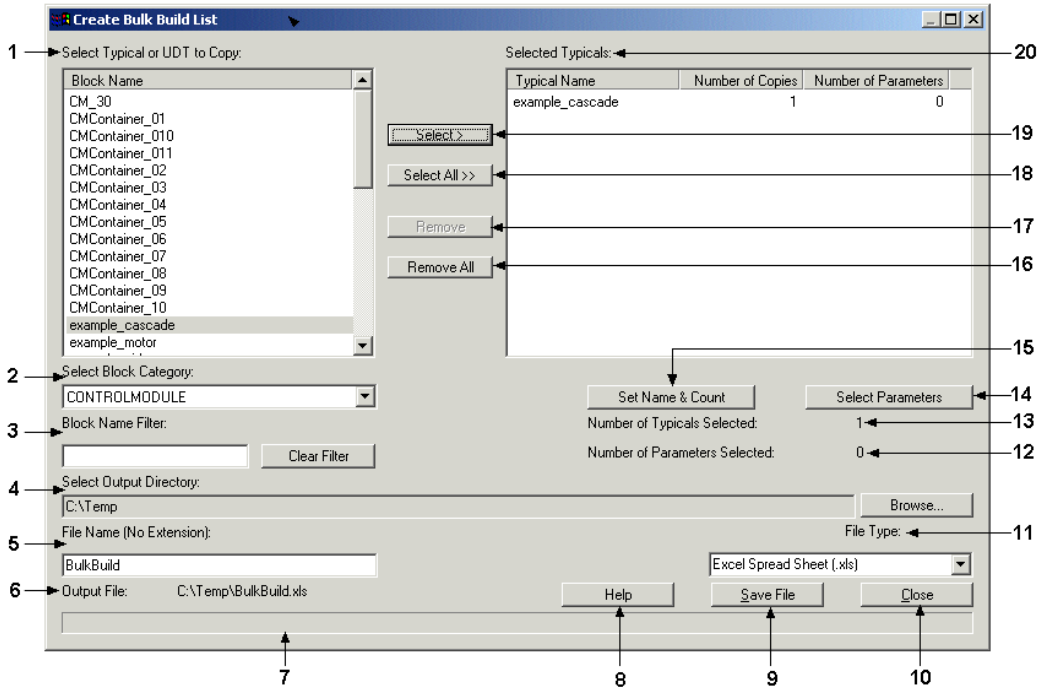
The Bulk Build utility is an advanced Control Builder configuration tool designed to streamline the process of duplicating existing control strategies. It lets you clone a typical object multiple times featuring automatic renaming and the ability to edit specified parameters through a generated spreadsheet.

This section includes information that you can reference for:

Topic	Link
Identifying functions on Create Bulk Build List dialog	Click here
Identifying functions on Set Name & Count dialog	Click here
Identifying functions on Select Parameters dialog	Click here
Creating a Bulk Build List file	Click here
Viewing Sample Spreadsheet File	Click here
Viewing Sample Access Database File	Click here
Identifying functions on Read Bulk Build File	Click here
Reading a Bulk Build File	Click here

Identifying functions on Create Bulk Build List dialog

The following illustration and callout table, identify the functions that are available on the **Create Bulk Build List** Dialog.



Create Bulk Build List dialog callouts

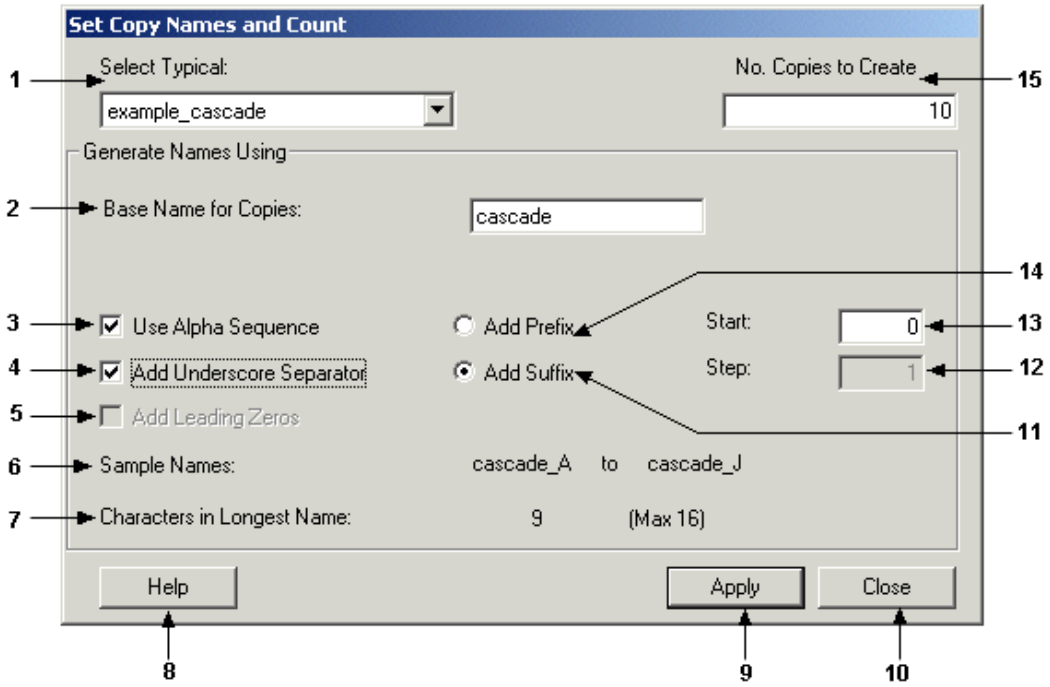
Callout	Function	Description
1	**Select Typical or UDT to Copy**	List box shows blocks in configured control strategies currently in the ERDB as filtered by the **Select Block Category** selection and the **Block Name Filter** entry on this dialog.

Callout	Function	Description
2	**Select Block Category** box	Click **Arrow** button to select desired module category filter for the **Select Typical or UDT to Copy** list. • CONTROLMODULE (default) • FIELDBUS • IOMODULE • SEQUENTIALCONTROLMODULE • USER
3	**Block Name Filter**	Key in desired characters to limit the blocks shown in the **Select Typical or UDT to Copy** list to only those whose name begins with the (case insensitive) characters entered in this field.
	Clear Filter Button	Click to clear the **Block Name Filter** field and restore the **Select Typical or UDT to Copy** list to its unfiltered condition.
4	**Select Output Directory**	Click the **Browse** button to launch **Browse** dialog and navigate to desired directory location for *saved* output file. The selected directory appears in the field and can only be edited through the **Browse** button.
5	**File Name (No Extension)**	Key in desired output file name without the file type extension in this field. The extension is automatically determined by the selection in the **File Type** box.
6	**Output File**	Read-only field shows directory location of the *saved* output file.
7	**Progress Bar**	Shows progress of the write operation to an output file during a *Save* function.
8	**Help** Button	Click to call up related topic in Knowledge Builder.
9	**Save File** Button	Click to save data by writing it to the output file. In the output file, a new row is created for each duplicate and/or contained block, and columns for all parameters. If a parameter is not selected for a particular block type or *Typical*, the corresponding entry in the output file will be blank. Otherwise, the current value from the original block (the one being duplicated) is written.

Callout	Function	Description
10	**Close** Button	Click to quit the function and close the dialog box.
11	**File Type** Box	Click the **Arrow** button to select desired output file type from list. • Excel Spread Sheet (.xls) (default) • Access Database (.mdb) • Delimited text (CSV)
12	**Number of Parameters Selected**	Read-only field shows number of Parameters Selected so far.
13	**Number of Typicals Selected**	Read-only field shows number of *Typicals* selected so far.
14	**Select Parameters** Button	Click to launch the **Select Parameters** Dialog. See *Identifying functions on Select Parameters dialog* for details.
15	**Same Name & Count** Button	Click to launch the **Same Name and Count** Dialog. See *Identifying functions on Set Name and Count dialog* for details.
16	**Remove All** Button	Click to remove all selected blocks from the **Selected Typical** list.
17	**Remove** Button	Click to remove selected blocks from the **Selected Typical** list.
18	**Select All** Button	Click to move all blocks selected in **Select Typical or UDT to Copy** list to **Selected Typicals** list.
19	**Select** Button	Click to move selected block in **Select Typical or UDT to Copy** list to **Selected Typicals** list.
20	**Selected Typicals**	List box shows: • Blocks that have been selected as Typicals by Name. • Number of copies (default is one) • Number of parameters (default is zero)

Identifying functions on Set Name and Count dialog

The following illustration and callout table identify the functions that are available on the **Set Copy Names and Count** dialog.



Set Name & Count dialog callouts

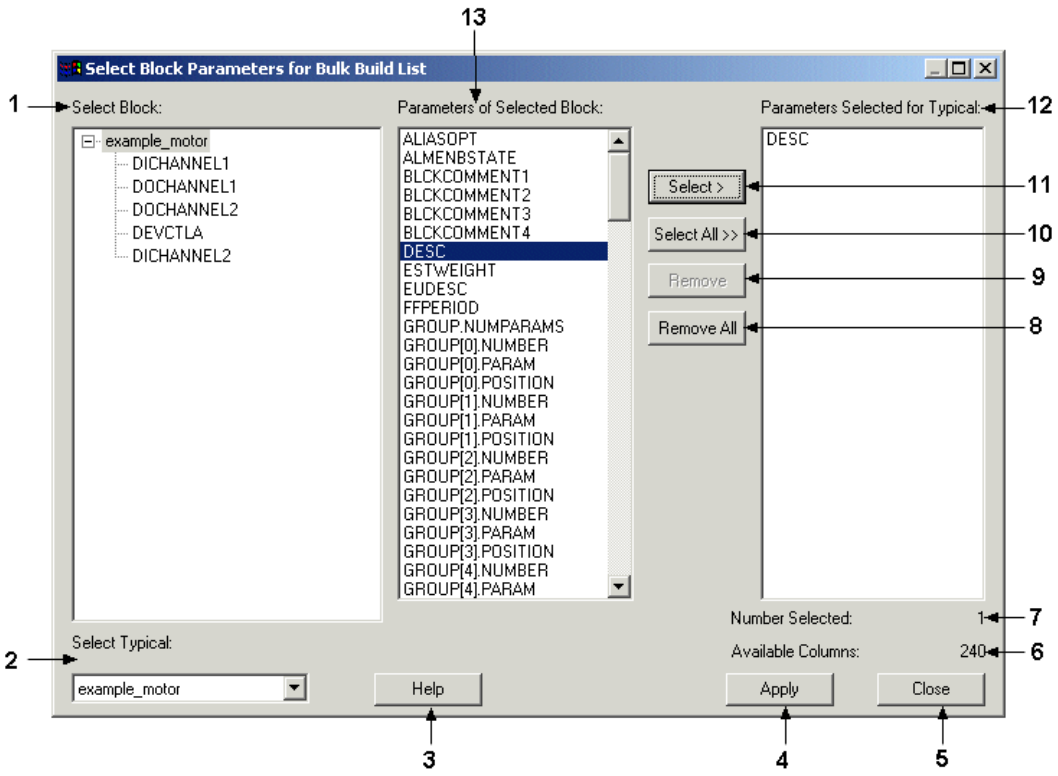
Callout	Function	Description
1	**Select Typical** box	Click the **Arrow** button to select the desired *Typical* for which the duplication count and naming is being set.
2	**Base Name for Copies**	Key in the desired base name for the copies to which the suffix or prefix will be appended. The default base name is the first 15 characters in the *Typical* name.

Callout	Function	Description
3	**Use Alpha Sequence** check box	Select this check box if you want the prefix or suffix for the base name generated as a sequence of upper case alphabetical characters starting at *A*. For example, *A to ZZZ*. Default is not selected.
4	**Add Underscore Separator** Check Box	Select this check box if you want the generated prefix or suffix separated from the base name by an underscore. Default is not selected.
5	**Add Leading Zeros** Check Box	Select this check box if you want numeric values used for the prefix or suffix to be formatted with leading zeros so all values are of an equal string length. If it is not selected, numeric strings will only be as long as is required for the value. Default is not selected.
6	**Sample Names**	Read-only field displays the first and last name to be generated based on current selections. The display updates as changes are made to the name generation components.
7	**Characters in Longest Name**	Read-only field displays the length of the longest proposed name. If the length of the longest proposed name exceeds 16 characters, the **Apply** button and the **Select Typical** box are disabled until you change naming selections so the longest proposed name does not exceed the 16 character maximum.
8	**Help** Button	Click to call up related topic in Knowledge Builder.
9	**Apply** Button	Click to save the current name generation selections for the specified *Typical*.
10	**Close** Button	Click to quit the function and close the dialog.
11	**Add Suffix** Option Button	Select this button if you want to generate a sequential suffix to construct the names for each copy. This button and the **Add Prefix** button are mutually exclusive.
12	**Step**	Key in a numeric value within the range of 0 to 100 to be used as the increment between suffix values. This field is unavailable if the **Use Alpha Sequence** check box is selected. In this case, the sequence is strictly Alphabetical with a one-character value step.

Callout	Function	Description
13	**Start**	Key in a numeric value within the range of 0 to 1000 to set the starting character for the sequence of suffixes or prefixes. Default is zero.
14	**Add Prefix** Option Button	Select this button if you want to generate a sequential prefix to construct the names for each copy. This button and the **Add Suffix** button are mutually exclusive.
15	**No. Copies to Create**	Key in the number of copies you want to make within the range of 1 to 5000.

Identifying functions on Select Parameters dialog

The following illustration and callout table identify the functions that are available on the **Select Block Parameters for Bulk Build List** dialog.



Select Parameters dialog callouts

Callout	Function	Description
1	**Select Block**	List box displays a tree view of the selected *Typical* with all its contained blocks expanded. Selection of a block in an expanded view populates the **Parameters of Selected Block** list with its associated parameters.
2	**Select Typical** Box	Click the **Arrow** button to select desired *Typical* to be shown in the **Select Block** list.
3	**Help** Button	Click to call up related topic in Knowledge Builder.
4	**Apply** Button	When available, click to save the parameter selections for the block selected in the tree view in the **Select Block** list.
5	**Close** Button	Click to quit the function and close the dialog.
6	**Available Columns**	Read-only field displays the number of parameter columns available for new selections. (The output is limited to 242 uniquely named parameters; the remaining 13 are used for fixed information about the *Typical* and its blocks.)
7	**Number Selected**	Read-only field displays the number of parameters in the **Parameters Selected for Typical** list.
8	**Remove All** Button	Click to remove all parameters the **Parameters Selected for Typical** list.
9	**Remove** Button	Click to remove the selected parameters from the **Parameters Selected for Typical** list.
10	**Select All** Button	Click to copy all parameters currently displayed in the **Parameters of Selected Block** list to the **Parameters Selected for Typical** list.
11	**Select** Button	Click to copy the current selections in the **Parameters of Selected Block** list to the **Parameters Selected for Typical** list. Duplicate parameters are ignored.
12	**Parameters Selected for Typical**	List shows the parameters selected for the current *Typical* (all blocks).

Callout	Function	Description
13	**Parameters of Selected Block**	List is populated with the parameters appropriate to the block type selected in the tree view in the **Select Block** list.

Creating Bulk Build List file

Prerequisites:

- You have launched Control Builder with a logon security level of Engineer or greater.
- You have already created control strategies in Control Builder that reside in the Engineering Repository Database (ERDB).
- You are familiar with using Control Builder to configure Experion control strategies.
- You understand which parameters are appropriate for specialization, and what values are acceptable for them.
- You understand the limitations and resource usage guidelines to properly assign objects to execution environments.

Considerations:

- You can select a maximum of 248 parameter names for all *Typicals*. While an Excel spreadsheet allows 255 columns, seven (7) columns are reserved for the duplication process.
- You can choose a subset of the parameters for each *Typical* object selected.These parameters are written to the output file with the current parameter value for subsequent editing.
- You do not need to install Microsoft Excel to generate an Excel spreadsheet, the Bulk Build utility uses Microsoft ActiveX Data Objects platform services to create an Excel spreadsheet output file.
- You can choose Excel Spreadsheet, Access Database, and Delimited text (CSV) as the output file type.
- The write file operations are done in the background so creating a bulk build file will not have significant performance impact. A file containing duplication specifications for several hundred objects will typically take no more than two to three minutes to create under normal conditions.

The following procedure outlines the general steps for creating a bulk build list file for example purposes only. You can easily adapt the steps to your specific application requirements.

Step	Action
1	On **Tools** menu, click **Bulk Build** > **Create Bulk Build List** to launch the utility.
2	In **Select Block Category** box, click **Arrow** button and select **CONTROLMODULE**.
3	*Optional:* Key in desired characters to limit/filter the block names that can appear in the **Select Typical or UDT to Copy** list or leave it blank to display all the control strategy data in the ERDB. For example, if you key in *ex*, only blocks whose name starts with *ex* will be displayed in the list. This entry is not case sensitive.
4	Click the **Browse** button to select new directory location for the output file in the **Output Directory Location** box or accept the default.
5	In **File Name (No Extension)** box, key in desired name to be used for the output file. For example, key in *NewBBF1*.
6	In **Select Typical or UDT to Copy** list, select the block you want to copy. For example, click *example_pid*.
7	Click the **Select** button to add seleted block to the **Selected Typicals** list.
8	Click **Set Name & Count** button to call up the dialog.
9	In **Set Name & Count** dialog, click the **Arrow** button in the **Select Typical** box to select desired Typical. For example, select *example_pid*.
10	In **No. Copies to Create** box, key in number of copies you want to make of this block. For example, key in *10*.
11	In **Base Name for copies** box, key in desire base name to be used for copies. For example, key in *expidcopy*.
12	Use remaining check boxes and option buttons to further define names for block copies. See *Identifying functions on Set Name and Count dialog* for more details.
13	Click **Apply** button to save the settings for the selected *Typical*.
14	Click **Close** button to close the dialog.
15	On **Create Bulk Build List** dialog, click **Select Parameters** button to call up the dialog.
16	On **Select Parameters** dialog, click **Arrow** button in **Select Typical** box to

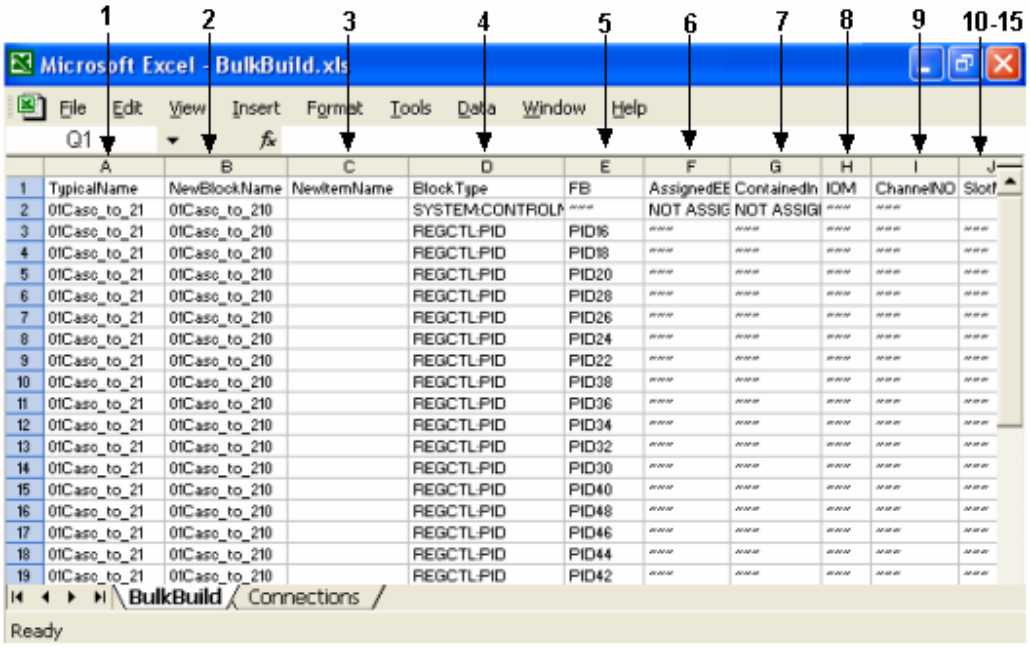
Step	Action
	select the block to appear in the **Select Block** tree view.
17	Select desired block in **Select Block** tree view to view its parameters in the **Parameters of Selected Block** list. See the *Identifying functions on Select Parameters dialog* for more details
18	In the **Parameters of Selected Block** list, select the parameters you want to include in the Bulk Build list for the *Typical* and click the **Select** button. Or, click the **Select All** button to copy all the parameters listed.
19	Click the **Apply** button to save the settings for the selected block of the selected *Typical*.
20	Click the **Close** button to close the dialog.
21	On the **Create Bulk Build List** dialog, click **Arrow** button in **File Type** box to select desired output file type. For example, select **Excel Spread Sheet (.xls)**.
22	Click the **Save File** button to initiate the writing of selected data to the output file. You can monitor progress of the write function in the Progress bar.
23	You can repeat this procedure as required for other *Typicals*.
24	Click the Close button to close the dialog.
25	This completes the procedure.

Viewing sample spreadsheet file

The following illustration and callout table identify the features that can be viewed on a typical spreadsheet file that is the output of the Create Bulk Build List utility.

Considerations

- The output spreadsheet contains a worksheet called BulkBuild. It always includes Columns A through N and the first row, which provides the *names* for the rest of the columns.
- Value indications can be
  - ~~~ - Three successive tilde characters mean that the parameter is not appropriate for the current block.
  - NOT ASSIGNED - Means that value is unassigned.
  - (Blank) - For Parameters, a blank cell will be interpreted to mean *same as Typical*. For fixed fields, it is treated as Not Assigned to avoid the errors that would result from duplicate assignment if the values in the typical were used.
- You can select up to 241 unique parameter names. If a parameter name is applicable to multiple templates; then selecting it for one template, results in it being selected for all.
- Parameter names that become column names can not include square brackets ([ ]), so the square brackets will be replaced with parentheses when written to the database.
- Parameter names that contain an embedded period (.) will have the period changed to a pound symbol (#) character when written to the database.
- Internal connections and connections to the blocks container (if specified) will be duplicated. Connections to external objects will be removed to avoid errors from duplicate connections. You should review all new blocks for needed connections.



Sample spreadsheet callouts

Callout	Column	Description
1	**A - TypicalName**	The name of the *Typical* being copied.
2	**B - NewBlockName**	The name of the new copied block.
3	**C - NewItemName**	The name for a new item.
4	**D - BlockType**	The full name of the System or User template for this block. Must match that in the *Typical*, should not be edited.
5	**E - FB**	The name of the contained Function Block (of the *Typical*), Not applicable for the top-level container block. Must match that in the *Typical*, should not be edited.
6	**F - AssignedEEFB**	The Execution Environment container block to which to assign this copy.
7	**G - ContainedIn**	The name of the Block that contains this block, for

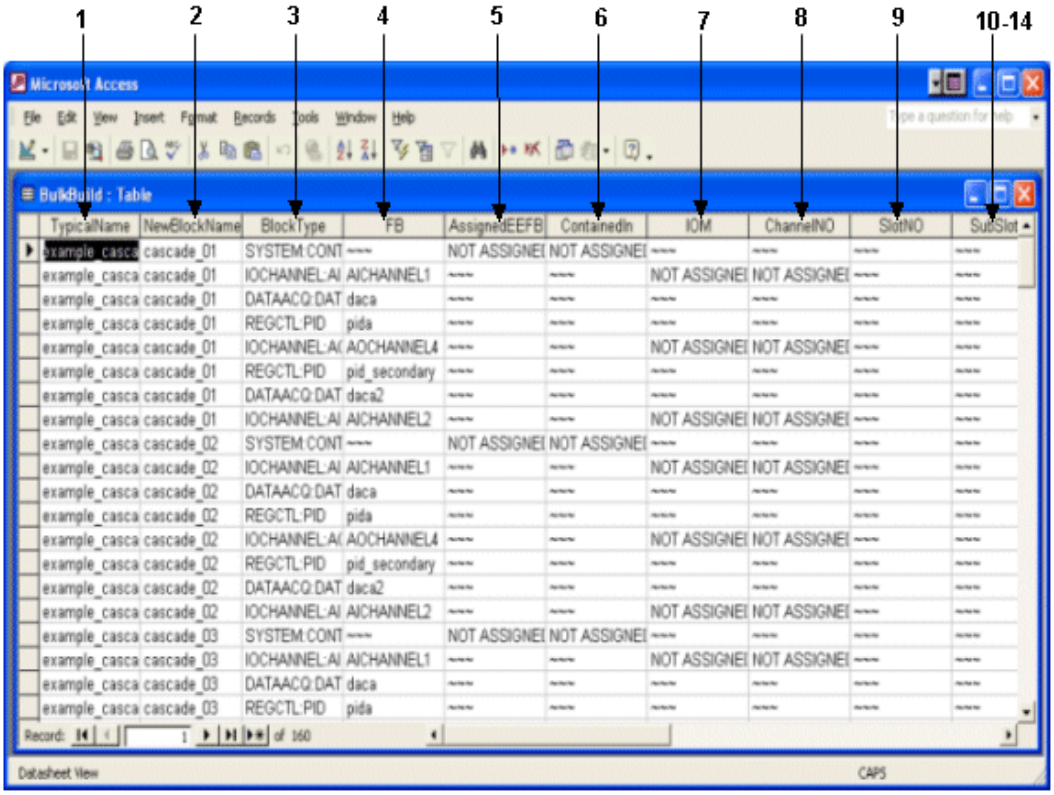
Callout	Column	Description
		hierarchical building. May be omitted, but will cause copies of blocks that are contained to be built as uncontained copies. If specified, the copies will be contained in the specified block (which can be part of the current bulk build) if it exists.
8	**H - IOM**	The I/O Module to which the copy is to be assigned.
9	**I - CHANNELNO**	The Channel number [or Box number for Experion Hiway Gateway (EHG)] within the I/O Module to assign this copy to.
10	**J - SlotNo**	The Slot Number (EHG only) for this device assignment.
Columns not shown		
11	**K - SubSlotNo**	The Sub-Slot Number (EHG only) for this device assignment.
12	**L - InSubSlot2**	The Input Sub-Slot Number 2 (EHG only) for this device assignment.
13	**M - OutSlotNo**	The Output Slot Number (EHG only) for this device assignment.
14	**N - OutSubSlotNo**	The Output Sub-Slot Number (EHG only) for this device assignment.
15	**O - OutSubSlot2**	The Output Sub-Slot 2 Number (EHG only) for this device assignment.
	Parameter Columns	The remaining columns in the spreadsheet are used to modify parameter values in the copies.

Viewing sample Access database file

The following illustration and callout table identify the features that can be viewed on a typical access database file that is the output of the Create Bulk Build List utility. The output is similar to the Excel Spreadsheet.

Considerations

- The output access database contains a table called BulkBuild. It always includes the first seven columns and the column names are the same as those used in the first row of the spreadsheet version.
- Value indications can be
  - ~~~ - Three successive tilde characters mean that the parameter is not appropriate for the current block.
  - NOT ASSIGNED - Means that value is unassigned.
  - (Blank) - Means that it will be interpreted to mean *same as Typical*.
- You can select up to 241 unique parameter names. If a parameter name is applicable to multiple templates; then selecting it for one template, results in it being selected for all.
- Parameter names that become column names can not include square brackets ([ ]), so the square brackets will be replaced with parentheses when written to the database.
- Parameter names that contain an embedded period (.) will have the period changed to a pound symbol (#) character when written to the database.



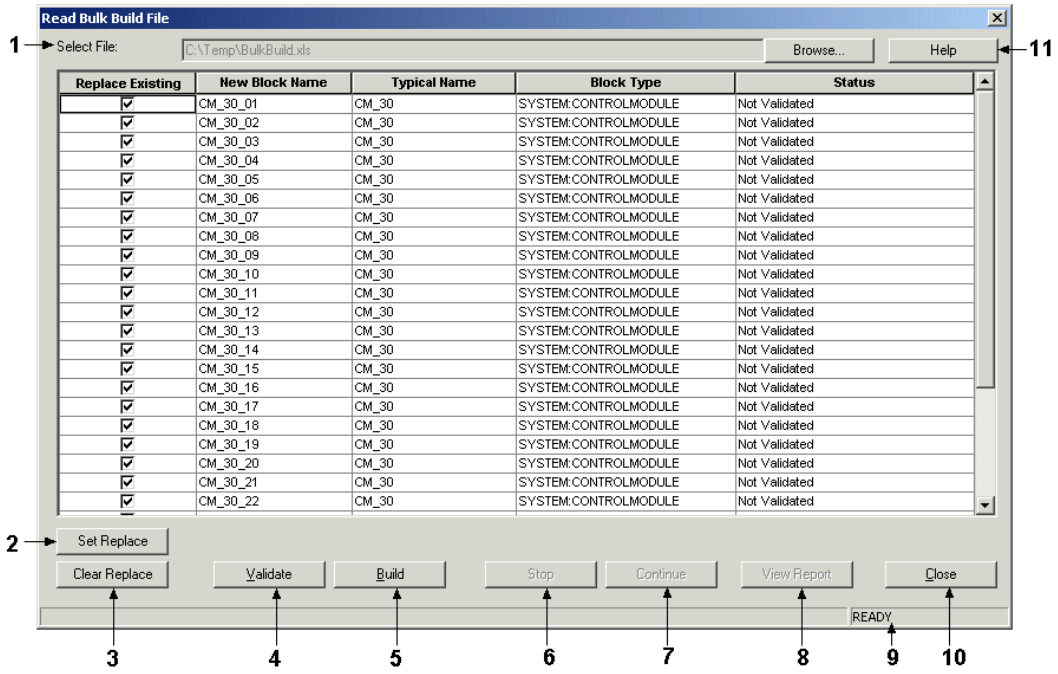
Sample Access database callouts

Callout	Column	Description
1	**TypicalName**	The name of the *Typical* being copied.
2	**NewBlockName**	The name of the new block copy.
3	**BlockType**	The full name of the System or User template for this block. Must match that in the *Typical*, should not be edited.
4	**FB**	The name of the contained Function Block (of the *Typical*), Not applicable for the top-level container block. Must match that in the *Typical*, should not be edited.
5	**ContainedIn**	The name of the Block that contains this block, for

Callout	Column	Description
		hierarchical building. May be omitted.
6	**AssignedEEFB**	The Execution Environment container block to which to assign this copy.
7	**IOM**	The I/O Module to which the copy is to be assigned.
8	**CHANNELNO**	The Channel number [or Box number for Experion Hiway Gateway (EHG)] within the I/O Module to assign this copy to.
9	**SlotNo**	The Slot Number (EHG only) for this device assignment.
Columns not shown		
10	**SubSlotNo**	The Sub-Slot Number (EHG only) for this device assignment.
11	**InSubSlot2**	The Input Sub-Slot Number 2 (EHG only) for this device assignment.
12	**OutSlotNo**	The Output Slot Number (EHG only) for this device assignment.
13	**OutSubSlotNo**	The Output Sub-Slot Number (EHG only) for this device assignment.
14	**OutSubSlot2**	The Output Sub-Slot 2 Number (EHG only) for this device assignment.
	Parameter Columns	The remaining columns in the table are used to modify parameter values in the copies.

Identifying functions on Read Bulk Build File

The following illustration and callout table identify the functions that are available on the **Read Bulk Build File** Dialog.



Read Bulk Build File dialog callouts

Callout	Function	Description
1	**Select File**	Shows the full path to the file selected to be read through the **Browse** button. The selected file will be validated for correct structure including required rows and columns present, and correct sheet or table name; and then read into the list view area below.

Callout	Function	Description
2	**Set Replace** Button	Click to set (check) all check boxes in the **Replace Existing** column at once. You can also individually select check boxes by row. When set, any blocks that exist with the same name as **New Block Name** will be overwritten with new information from the *Typical* and the spreadsheet.
3	**Clear Replace** Button	Click to clear (blank) all check boxes in the **Replace Existing** column at once. You can also individually clear check boxes by row.
4	**Validate** Button	Click to have the data read in verified against the ERDB to insure that the *Typicals* being copied are present and are structurally consistent with the spreadsheet (no missing basic blocks). The results are updated in the **Status** column of the list. This button is only available after a valid selected file is read. Once this function has been completed, the **View Report** button is available. While the validate function is executing, the **Status** bar will display the text *Validating File* and a progress bar will be displayed in the right pane of the **Status** bar.
5	**Build** Button	Click to build the duplicates specified in the spreadsheet that were not excluded due to errors in the validate phase. This button is only available after a valid selected file is read. If the read file has not been validated by clicking the **Validate** Button first, the file will be validated before the build step is initiated for all valid items. The **Status** bar first displays *Ordering Build*, and then *Building blocks*. During the *Building Objects* phase a progress bar will be displayed in the right pane of the **Status** bar.
6	**Stop** Button	Click to stop the validation or build process at the end of the next complete step. This button is only available after a validation or build has started.
7	**Continue** Button	Click to resume a stopped validation or build process sequence at the next step. This is only applicable after a **Stop** button action.

Callout	Function	Description
8	**View Report** Button	Click to display the log file created during the validation and build steps in the Notepad application. The log file contains entries for each object, showing successful creation, skipped objects and all errors encountered in both phases. The log file is created in the standard Experion error log directory *C:\ErrorHandling*.
9	**Status Bar**	Shows the current operational state in the right pane and the current progress during validate and build in the left pane.
10	**Cancel** Button	Click to cancel the operation and close the dialog. If a validate or build step is in progress, a prompt asks you to confirm the action and the process will be aborted at the next possible step.
11	**Help** Button	Click to call up related topic in Knowledge Builder.

Reading Bulk Build File

Prerequisites:

- You have launched Control Builder with a logon security level of Engineer or greater.
- You have already created a valid Bulk Build list file in an Excel Spreadsheet, Access Database, or text file format using the Bulk Build List dialog. Or, you have created a spreadsheet using Microsoft Office 97 or later that conforms to the expected format and the named blocks to be duplicated actually exist in the current Engineering Repository Database (ERDB).
- You are familiar with using Control Builder to configure Experion control strategies.
- You understand which parameters are appropriate for specialization, and what values are acceptable for them.
- You understand the limitations and resource usage guidelines to properly assign objects to execution environments.

Considerations:

- Bulk Build will not modify Qualification and Version Control System (QVCS) parameters for new objects; they will be set to the normal defaults for a new object. You must use normal QVCS procedures to modify the Qualification State of the new objects. Overwrite of an existing object on systems with full, licensed QVCS active will require manual check out and check in.
- The *validate* and *build* operations are done in the background so reading a bulk build file will not have significant performance impact.

The following procedure outlines the general steps for reading a bulk build list file for example purposes only. You can easily adapt the steps to your specific application requirements.

Step	Action
1	On **Tools** menu, click **Bulk Build** > **Read Bulk Build List** to launch the utility.
2	Click the **Browse** button to navigate to the directory location that contains the Bulk Build List file to be read and select it so it appears in the **Select File** box.

Step	Action
3	Check that selected file contents is displayed in the list view area with one complete duplicated object per row.
4	Click the **Set Replace** button to select all the check boxes in the **Replace Existing** column. This means any blocks that exist with the same name as **New Block Name** in the ERDB will be overwritten with new information from the *Typical* and the spreadsheet. If **Replacing Existing** check box is not selected and an object with the same name exists in the database, an error message is generated and the existing object is not overwritten in the database.
	**TIP** You may chose to skip Steps 5 and 6, and go to Step 7 to just click the **Build** button first. In this case, the validation is done immediately before the build process starts.
5	Click the **Validate** button. Check the **Status** bar to monitor progress of the validation process.
6	Check the **Status** column to review the validation status of each duplicate. The validation status can be either: • **Not Validated (initial status)** • **Typical Does Not exist** • **Block Type mismatch in typical** • **Embedded Block Not in Typical** • **Validated** • **Validated - Replace Existing Required** Only those duplicates with a **Validated or Validated - Replace Existing Required** status will be considered for a build (replace existing must be checked for the latter).
7	Once validation is completed, click the **Build** button. Check the **Status** bar to monitor the progress of the build process.
8	Check the Status column to review the build status. It can be one of these: • **Build Ok** • **Block Exists -Not Replaced**

Step	Action
	• **Build Errors** **Assignment Errors**
9	Once the build status is checked, click the **View Report** button to view the error log in the Notepad application and check for skipped objects and any errors encountered during the validate and build phases.
10	Close Notepad and click the **Cancel** button to close the dialog.
11	Check duplicates in the **Project** tab of Control Builder and make any configuration changes as you normally would.
12	This completes the procedure.

On-line monitoring using Control Builder

This section includes information that you can reference for:

Topic	Link
Initial activation order guidelines	Click here
Activating the CEE	Click here
Setting the CEE inactive	Click here
Setting I/O active	Click here
Setting I/O inactive	Click here
Changing parameters while monitoring	Click here
Operator actions versus IOP outputs	Click here
Interacting with ACE	Click here

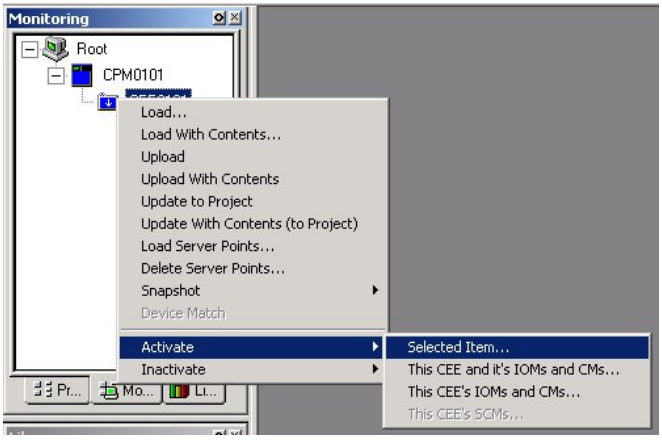
Initial activation order guidelines

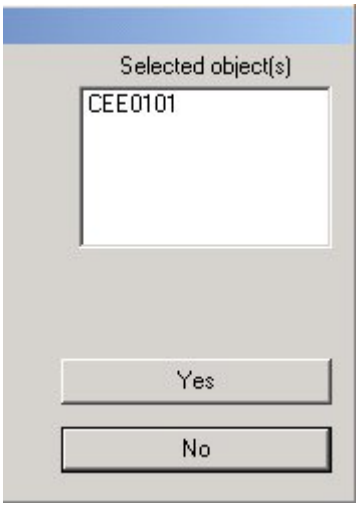
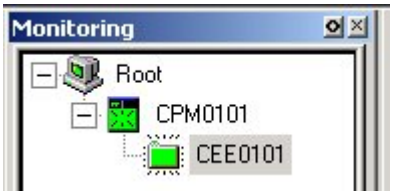
Make the initial activation of control strategy components from the Monitoring tab in the following suggested order to minimize possible bad data generated alarms.

Order	Component
1	Control environment components such as: CPM/CEE ACE/CEE SCE
2	Input/Output Processors (IOPs)
3	PMIO I/O Channels
4	Fieldbus contained function blocks
5	Fieldbus device resident blocks
6	Input/Output Modules (IOMs)
7	Control Modules (CMs) and/or Sequential Control Modules (SCMs)

Activating the CEE

Follow the steps in the table below to activate the CEE.

Step	Action	Result
1	From the Monitoring Tree, right-click the CEE.	
2	Click Activate and then the desired selection as follows to activate multiple items: • Selected Item … • This CEE and its IOMs and CMs • This CEE's IOMs and CMs … • This CEE's SCMs … 	

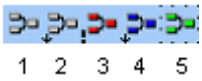
Step	Action	Result
3	Click YES from the pop-up window to set the selected item active.  Alternate methods: • Click the Toggle State toolbar button  to set the selected item active. • Select Operate -> Activate and then the desired selection as follows to activate multiple items: – Selected Item … – This CEE and its IOMs and CMs – This CEE's IOMs and CMs … – This CEE's SCMs …	The CEE turns green on the Monitoring Tree. • Remember that the CEE must be active for anything assigned to the CEE to work. Even if a CM is active, it won't do anything unless the CEE is also active. 

Refer to the table below for Monitoring Tree objects status information.

CPM		
	States	***Corresponding enum values***
	c) Project Tree	
	d) Loaded, not monitoring	
	e) Communication Failure	
	f) Not Used	
	g) Active	PCMSTATE = CEERUN, CEEMIX, BACKUP
	h) Failed	PCMSTATE = FAILED, OFFNET, TESTING, BOOTING
	i) Not Loaded	PCMSTATE = ALIVE, LOADING, PIREADY, NOTLOADED
	j) Loaded	PCMSTATE = NOCEE, CEEIDLE
Primary CPM		
Secondary CPM		
	States	***Corresponding enum values***
	k) Project Tree	
	l) Loaded, not monitoring	
	m) Communication Failure	

	n) Not Used	
	o) Active (in Synch)	PCMSTATE = CEERUN, CEEMIX, BACKUP and RDNSYNCSTATE = kCpmSyncMaintenance
	p) Failed	PCMSTATE = FAILED, OFFNET, TESTING, BOOTING
	q) Not Loaded	PCMSTATE = ALIVE, LOADING, PIREADY, NOTLOADED
	r) Loaded	PCMSTATE = NOCEE, CEEIDLE
	s) Not Synched	PCMSTATE = CEERUN, CEEMIX, BACKUP and RDNSYNCSTATE = kCpmNotInSync, kCpmPartnerVisible, kCpmSyncInProgress
	t) Standby	PCMSTATE = CEERUN, CEEMIX, BACKUP and RDNSYNCSTATE = kCpmSyncStandby
Primary RM		
Secondary RM		
	States	***Corresponding enum values***
	a) Project Tree	

	b) Loaded, not monitoring	
	c) Communication Failure	
	d) Not Used	
	e) Active (in Synch)	GENSTATE= kMGenStateOK and RDNSYNCSTATE = kRMReadiStateSynchronized
	f) Failed	GENSTATE= kRMGenStateUndefined, kRMGenStateStartup, kRMGenStateReload, kRMGenStateFault
	g) Not Used	
	h) Not Used	
	i) Not Synched	GENSTATE= kMGenStateOK and RDNSYNCSTATE = KRMReadiStateUndefined, kRMReadiStateNopartner, kRMReadiStateDisqualified, kRMReadiStateSynchronizing
	j) Standby	GENSTATE= kGenStateOK and READINESS= kRMReadiStateStandby
CEE	 	
	States	***Corresponding enum values***

	k) Project Tree	
	l) Loaded, not monitoring	
	m) Communication Failure	
	n) InActive	CEESTATE = IDLE
	o) Active	CEESTATE = RUN
	p) Configuration Error	CEESTATE = FAIL, NOT_LOADED
CM	 1 2 3 4 5	
	States	***Corresponding enum values***
	q) Project Tree	
	r) Loaded, not monitoring	
	s) Communication Failure	
	t) Inactive	EXECSTATE = Inactive
	u) Active	EXECSTATE = Active
SCM	 1 2 3 4 5 6 7	
	States	***Corresponding enum values***
	v) Project Tree	
	w) Loaded, not monitoring	
	x) Communication Failure	

	y) Inactive	STATE = kStsNulling, kStsNull, kStsNone, kStsEditing
	z) Active	STATE = KStsEdited, kStsRunning, kStsComplete, kStsChecking, kStsIdle, kStsInterrupting, kStsInterrupted, kStsRestarting, kStsRestarted, kStsHolding, kStsHeld, kStsStopping, kStsStopped, kStsAborting, kStsAborted, kStsCommErr
	aa) Not Used	
	bb) Configuration Error	ConfigSts != 0
Basic blocks		
	States	***Corresponding enum values***
	cc) Project Tree	
	dd) Communication Failure	
	ee) Configuration Error	ConfigSts != 0
IOM		
	States	***Corresponding enum values***
	ff) Project Tree	
	gg) Loaded, not monitoring	
	hh) Communication Failure	
	ii) Inactive	IOMSTATE = kIomInactive, kIomStart
	jj) Active	IOMSTATE = kIomRunning, kIomStub

	kk) Failed	IOMSTATE = kIomFail, kIomStandBy, kIom6
FIM	 1 2 3 4 5 6 7 8	
	States	***Corresponding enum values***
	ll) Project Tree	
	mm)Loaded, not monitoring	
	nn) Communication Failure	
	oo) Not Used	
	pp) Active	FIMSTATE = CEERUN, CEEMIX, BACKUP
	qq) Failed	FIMSTATE = FAILED, OFFNET, TESTING, BOOTING
	rr) Not Loaded	FIMSTATE = ALIVE, LOADING, PIREADY, NOTLOADED
	ss) Loaded	FIMSTATE = NOCEE, CEEIDLE
LINK	 1 2 3 4 5 6 7 8	
	States	***Corresponding enum values***
	tt) Project Tree	
	uu) Loaded, not monitoring	
	vv) Communication Failure	
	ww) Inactive	STATE = kDown

	xx) Active	STATE = kOnline
	yy) Failed	STATE = KFmwLoadErr, kCommInitErr, kDbInitErr
	zz) Active With UCD	STATE = KOnline And Number of UCDs > 0
	aaa)Going Active	STATE = KFmwLoad, kDbInit, kCommInit
Uncomm-issioned Device		
DEVICE		
	States	***Corresponding enum values***
	bbb) Project Tree	
	ccc) Loaded, not monitoring	
	ddd)Communication Failure	
	eee)Not Used	
	fff) Active	DEVSTATE = kDevOnline
	ggg)Failed	DEVSTATE = kDevOffline
Resource block		
Transducer block		

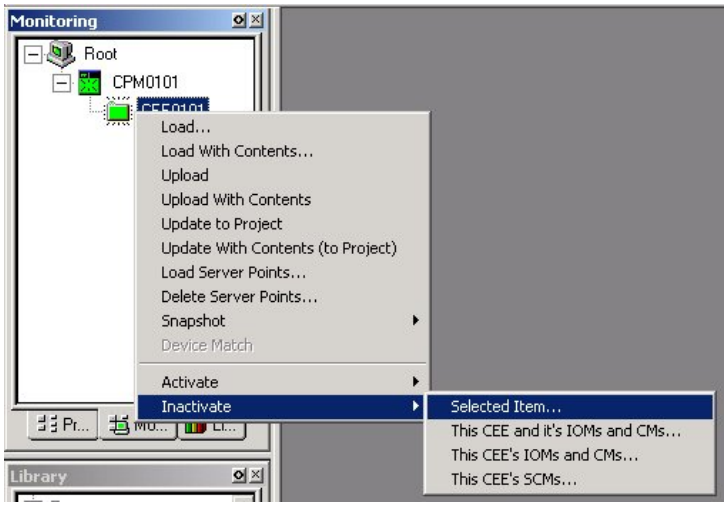
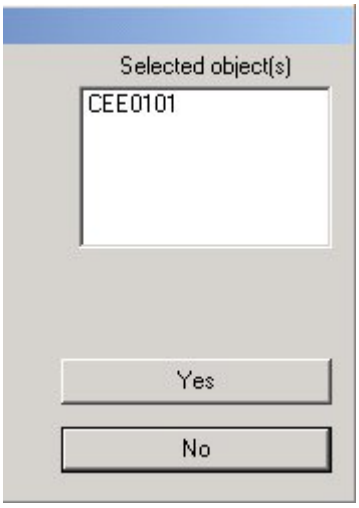
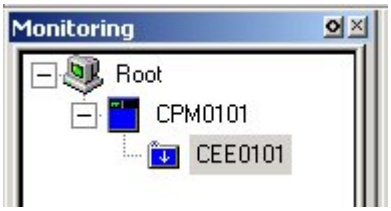
FF Function block		
	States	***Corresponding enum values***
	hhh) Project Tree	
	iii) Loaded, not monitoring	
	jjj) Communication Failure	
	kkk) Inactive	MODE.ACTUAL = kFFModeOOS
	lll) Active	MODE.ACTUAL = KFFModeNORMAL, kFFModeIMAN, kFFModeLO, kFFModeMAN, kFFModeAUTO, kFFModeCAS, kFFModeRCAS, kFFModeROUT

Setting the CEE inactive

Follow the steps in the table below to set the CEE inactive.

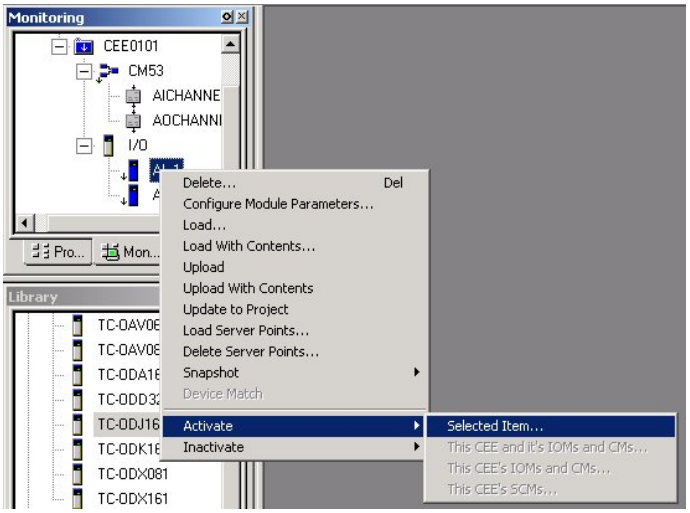
Step	Action	Result
1	From the Monitoring Tree, right-click the CEE.	

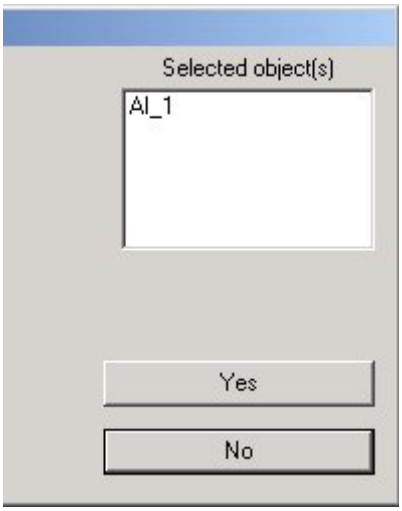
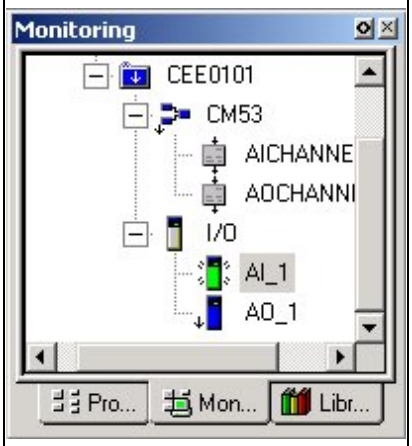
Step	Action	Result
2	Click Inactivate and then the desired selection as follows to activate multiple items: • Selected Item … • This CEE and its IOMs and CMs • This CEE's IOMs and CMs … • This CEE's SCMs … Alternate methods: • Click the Toggle State  toolbar button to set the selected item inactive. • Select Operate -> Inactivate and then the desired selection as follows to make multiple items inactive: – Selected Item … – This CEE and its IOMs and CMs – This CEE's IOMs and CMs … – This CEE's SCMs …	

Step	Action	Result
		
3	Click YES from the pop-up window to set the selected item inactive. 	The CEE turns from green to blue on the Monitoring Tree. 

Setting I/O active

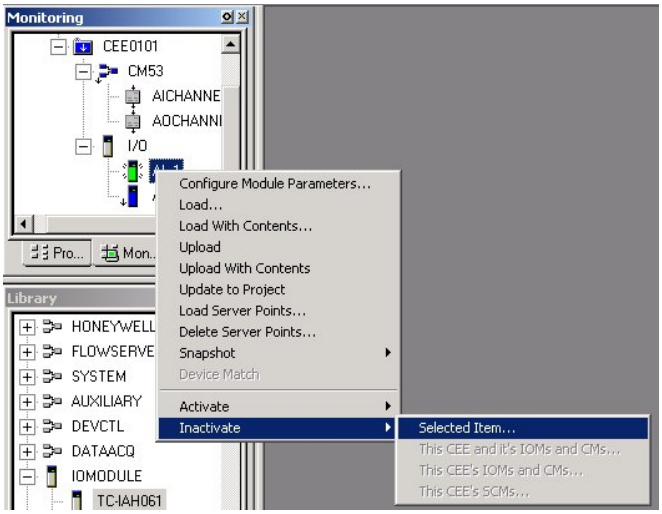
Follow the steps in the table below to set I/O to an active state.

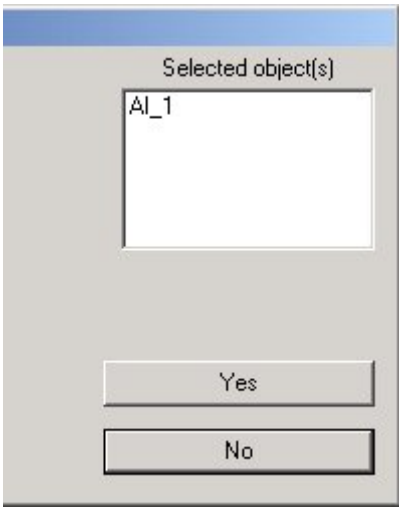
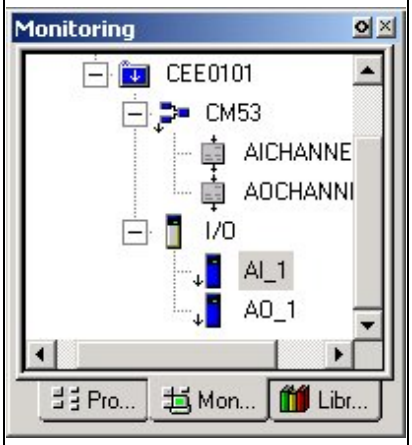
Step	Action	Result
1	From the Monitoring Tree, right-click the desired I/O block.	
2	Click Activate. 	

Step	Action	Result
3	Click Yes from the pop-up window to set the selected item active.  Note: IOMs can also be activated through the server database (that is, an IOM status point).	The I/O block turns green on the Monitoring Tree. 

Setting I/O inactive

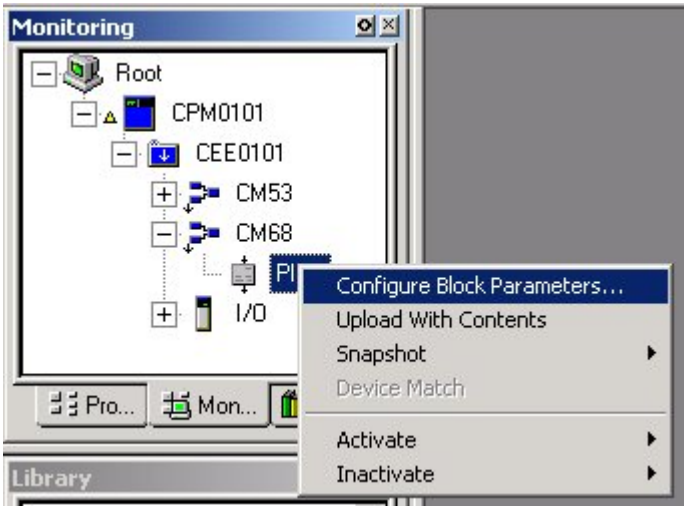
Follow the steps in the table below to set the I/O inactive.

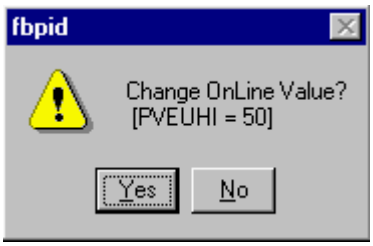
Step	Action	Result
1	From the Monitoring Tree, right-click the desired I/O block.	
2	Click Inactivate. Alternate method: Click the Toggle State  toolbar button to set the selected item inactive.	
		

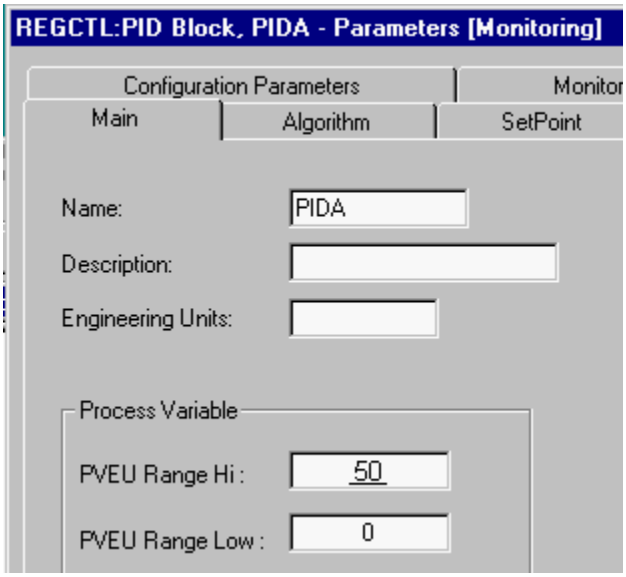
Step	Action	Result
3	Click Yes from the pop-up window to set the selected item inactive.  Note: IOMs can also be activated through the server database (that is, an IOM status point).	The I/O block turns from green to blue on the Monitoring Tree. 

Changing parameters while monitoring

Follow the steps in the table below to change parameters while monitoring a strategy. Note that the following is one specific example of changing parameters while monitoring using a PID block.

Step	Action
1	Click on the Monitoring Tree.
2	Right-click on the desired block  In this example, the block PIDA was selected. Note: CEE must be idle or FB must be inactive in order for some parameter changes to be allowed. For example, you can change the Enable Alarming Option on-line when both the CEE and function block are active.

Step	Action
3	Click on Configure Block Parameters, then enter the parameter changes. When the OK button is clicked, or the cursor is placed in any other field a confirm change dialog box appears. Confirm the change dialog as YES, if you want to write the new value to the controller, otherwise select No. Tip: Press the SHIFT-TAB keys to change the focus between the Yes and No buttons. Note: Any parameters changed while monitoring are only written to the controller; they are not written to the ERDB. 

Step	Action
4	In this example, the parameter PVEU Range Hi was changed to 50. The changed parameter will now appear underlined. 

Operator actions versus IOP outputs

The following table summarizes the affects that given operator actions can have on IOP outputs and/or related functions.

If operator. . .	Then, this occurs. . .	And, indications are. . .
Changes execution state (IOMSTATE) of an output IOP to **IDLE**	• For all channels, field output terminals **hold the value** applied at the time of idling. • Initialization Request (INITREQ) is asserted on all associated Output Channel Blocks. • No fault alarms should occur in Server (communication with the module is not disrupted). • The Detail Display shows **IDLE** state.	• IOP icon turns blue in Monitor tab • Detail Display shows **IDLE** state • No change in the IOP status LED display.
Inactiviates the Point Execution State (PTEXECST) of an Output Channel	• For all channels, field output terminals **hold the value** applied at the time of idling. The exception is DO channels executing **ONPULSE**/**OFFPULSE** commands. Inactivating the PTEXECST will not affect the output terminals, the pulse will complete at the specified time. • Initialization Request (INITREQ) is asserted on the IOC block. • No fault alarms should occur in Server (communication with the module is not disrupted).	• IOP icon turns blue in Monitor tab • No change in the IOP status LED display.

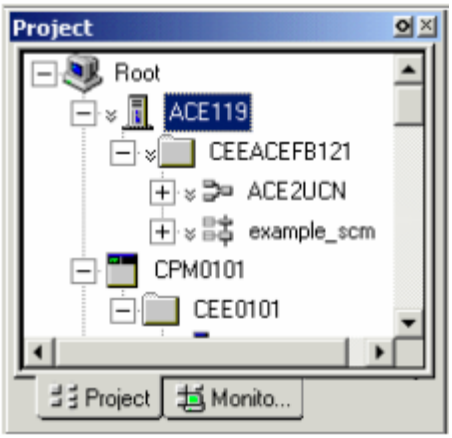
If operator. . .	Then, this occurs. . .	And, indications are. . .
Inactivates a Control Module containing an output channel	• For Output Channel block, execution is **NOT** affected. • No fault alarms should occur in Server (communication with the module is not disrupted).	• Control Module icon turns blue in Monitor tab. • I/O Channel icon remains green in Monitor tab. • The Detail Display shows **INACTIVE** state for the CM. • No change in the IOP status LED display.
Inactivates CEE containing PMIO output block	• No affect on output channel block execution. • No fault alarms should occur in Server (communication with the module is not disrupted).	• CEE icon turns blue in Monitor tab. • I/O Channel icon remains green in Monitor tab. • The Detail Display shows **IDLE** state for the CEE. • No change in the IOP status LED display.
Deletes IOP from Monitor tab	• IOP is currently in its IDLE state. • The IOP has no associated IOC blocks loaded (contained in CMs and appearing in the Monitor tab).	• IOP icon does not appear in Monitor tab.
Deletes an IOP from Project tab	• IOP is not loaded or has been deleted from Monitor tab. • IOP has no associated I/O Channel blocks contained in Control Modules.	• IOP icon does not appear in Project or Monitor tab.
Deletes a Control Module containing PMIO Output Channel blocks from Monitor tab	• I/O Channel block's Point Execution State must be **INACTIVE** or IOP must be in its **IDLE** state. • Sets all outputs to **UNPOWERED**.	• CM icon does not appear in Monitor tab.

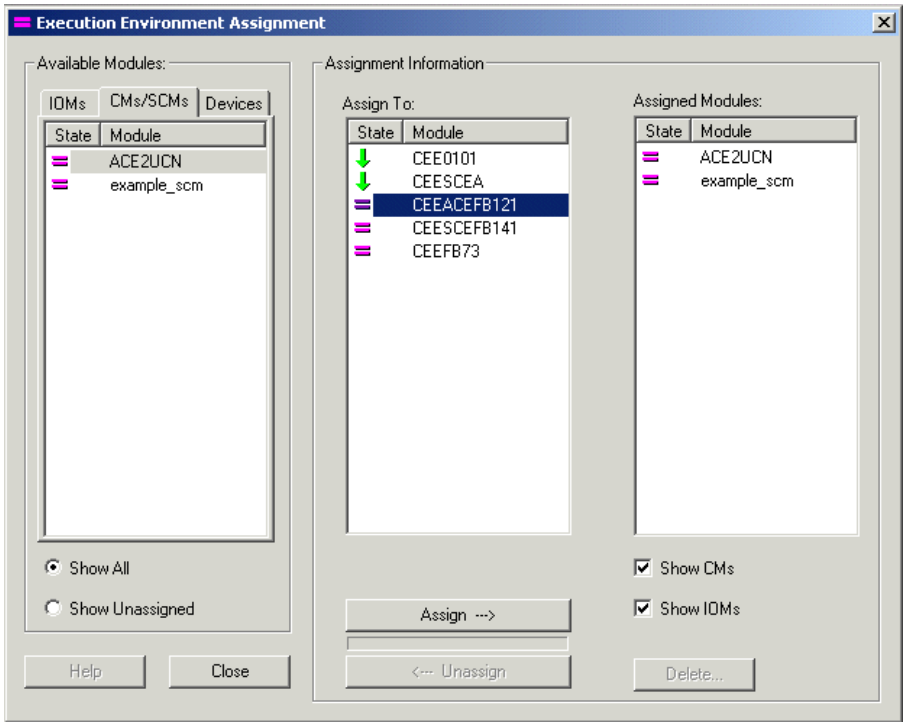
If operator. . .	Then, this occurs. . .	And, indications are. . .
Deletes PMIO output Channel block from CM in Project tab and re-loads CM to controller	• Output block's Point Execution State must be set to **INACTIVE**. • Output is set to **UNPOWERED**	• Output Channel icon does not appear in CM.

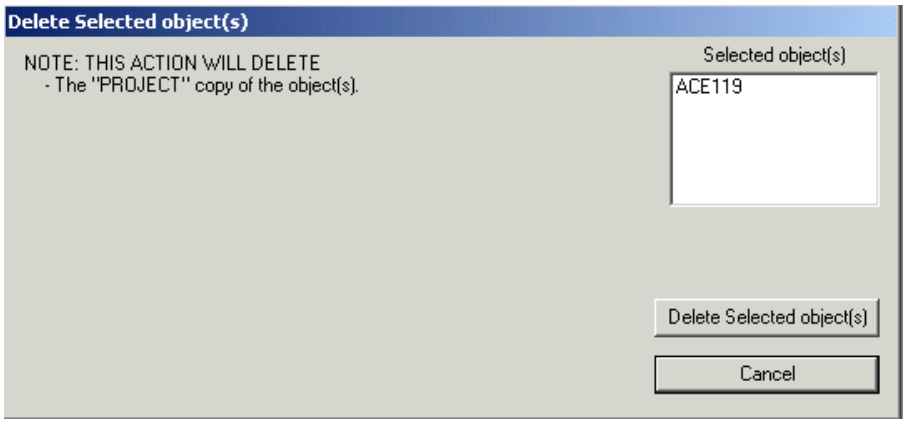
Deleting ACE/CEEACE block

Use the following procedure as a general guide for deleting a loaded ACE/CEEACE block from the Control Strategy. This procedure assumes that Control Builder is running with tree views open. The illustrations used in this procedure are for example purposes only.

Step	Action	Result
1	In Monitor Mode, open root directory for ACE/CEEACE.	Exposes contents of the CEEACE.
2	Right-click the CEEACE block and select Inactivate->Selected CEE(s), IOMs, CMs, Applicable Function Blocks from the shortcut menu.	Inactivates all components including the CEEACE. Block icons turn blue.
	TIP If ACE/CEEACE and its assigned components have been loaded, you must first put the CEEACE in its Idle mode and delete all of its components in the Monitor mode before you can delete them from the Project mode.	
3	Select components contained in CEEACE and click  delete button in the tool bar.	Calls up Delete Selected Objects dialog.
4	Click the Deleted Selected Object(s) button.	Initiates the delete function and progress dialog tracks status until complete.
5	Click the ACE block and click the delete button in the tool bar.	Calls up Delete Selected Objects dialog.
6	Click the Delete Selected object(s) button.	Initiates the delete function and progress dialog tracks status until complete.

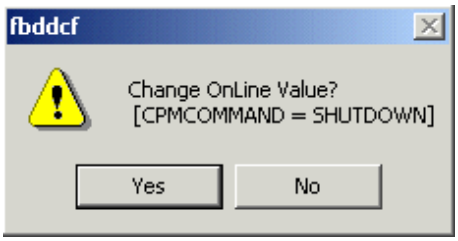
Step	Action	Result
		
7	Click the Project tab.	Calls up the Project view. 
8	With ACE selected, click  assign button in the tool bar.	Opens the Execution Environment Assignment dialog box.

Step	Action	Result
		
9	Click module assigned to ACE in Available Modules list, select CEEACE in Assign To list, select all the modules listed in the Assigned Modules list and click the Unassign button.	Unassigns modules from CEEACE.
10	Click the Close button.	Closes dialog box and returns to Project view.
11	With ACE selected, click  delete button in tool bar.	Calls up Delete Selected Objects dialog.

Step	Action	Result
		
12	Click the Delete Selected object(s) button.	Initiates the delete function and progress dialog tracks status until complete. 
13	This completes the deleting ACE/CEEACE procedure.	Stops the ace.exe on the ACE node, but the Control Data Access for supervisory platform (CDA-sp) continues to run.

Issuing a shutdown command and using Checkpoint to restore ACE

Use the following procedure as a general guide for issuing a Shutdown command to a loaded ACE/CEEACE block through the Monitor mode in Control Builder. This procedure assumes that Control Builder is running with tree views open. The illustrations used in this procedure are for example purposes only.

Step	Action	Result
1	In Monitor mode, right-click the CEEACE block and select Inactivate->Selected CEE(s), IOMs, CMs, Applicable Function Blocks from the shortcut menu.	Calls up Change State dialog.
2	Click the Yes button to continue.	Inactivates all components including the CEESCE. Block icons turn blue.
3	Right-click the ACE icon and select Configure Module Parameters.	Calls up the ACE Block configuration form.
4	On Main tab, click down arrow button in ACE Command field and select shutdown.	Calls up warning prompt for confirmation of online change. 
5	Click the Yes button.	Initiates shutdown of ACE, ACE turns yellow and other icons turn red. (The ace.exe stops, but the Control Data Access for supervisory platform (CDA-sp) continues to run. Use Administrative Tools in Windows Control Panel to stop the CDA-sp service on the ACE node.)
6	Right click ACE icon and select Restore from Checkpoint from shortcut menu.	Opens Restore from Checkpoint dialog.
7	Select node to be restored and click the Restore button.	Restores ACE data using the selected checkpoint files and icons turn blue. (This Step assumes that the CDA-sp service is running.)

Step	Action	Result
8	Right-click CEESCE icon and select Activate->Selected CEE(s), and its IOMs, CMs and Applicable Function Blocks from the shortcut menu.	Calls up Change State dialog.
9	Click the Yes button to continue.	Activates components and icons turn green.
10	This completes the Shutdown Command and checkpoint recovery procedure.	

Recovering from power failure

Use the following procedure as a general guide for recovering ACE/CEEACE operation after a power failure. This procedure assumes that Control Builder is running with tree views open. The illustrations used in this procedure are for example purposes only.



TIP

You can use this same general procedure to recover from an ACE application failure.

Step	Action	Result
1	Power to ACE node is lost.	Generates loss of ACE communications event and ACE/CEEACE and contained component block icons turn red in Monitor Mode of Control Builder on Server.
2	Restore power to ACE node.	Wait for ACE node to return to operation (Boot).
3	In Monitor mode on Control Builder, right-click the ACE block icon and select. Restore from Checkpoint from shortcut menu.	Opens Restore from Checkpoint dialog.
4	Select node to be restored and click the Restore button.	Restores ACE data using the selected checkpoint files and icons turn blue.
5	Right-click CEEACE icon and select Activate->Selected CEE(s), and its IOMs, CMs and Applicable Function Blocks from the shortcut menu.	Calls up Change State dialog.
6	Click the Yes button to continue.	Activates components and icons turn green.
7	This completes the power failure recovery procedure.	

Control Builder System Administration

Setting system preferences

Use the following procedure to customize the look of your Control Drawing and enable or disable selected system functions by using the System Preferences dialog.

Enabling selected system functions

Prerequisites:

- Control Builder is running
- Tree windows are open

Considerations:

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



ATTENTION

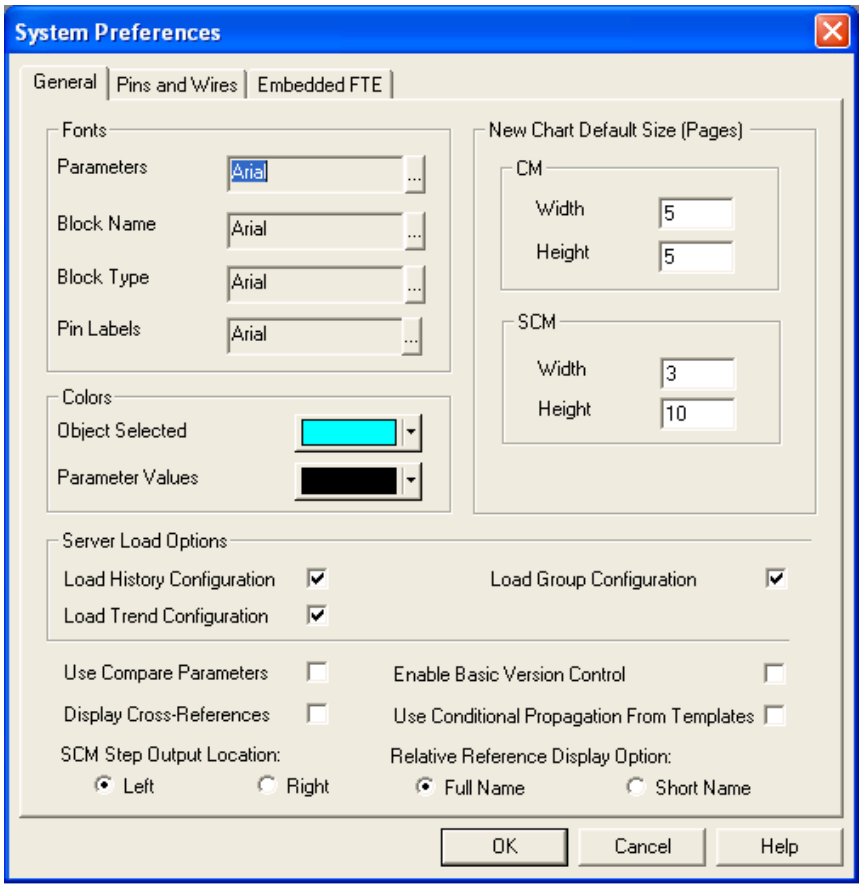
- User preferences are stored in the registry and are intended for use by the current user.
- System preferences are stored in the database and are intended for use by the entire system.
- Both Checkpoint configuration and operation permissions can be set to assist in saving and restoring operation and configuration data associated with a control strategy.

  Use the following links to access Checkpoint information:

  Error! Reference source not found.

  Error! Reference source not found.

To establish General properites using System Preferences dialog

Step	Action
1	Click Tools-> System Preferences to call up the System Preferences dialog.
	
2	On the General Tab, click the Ellipsis button [...] to right of the Parameters entry field to call up the Font dialog box.
3	Select the desired font typeface, style, size and effects to be used for parameter data in Control Builder. Click OK to close the Font dialog.

Step	Action
4	Repeat Steps 2 and 3 to make desired Font selections for the Block Name, Block Type, and Pin Labels data in Control Builder.
5	Click the down arrow  to the right of the Object Selected entry field to open the color palette. Click the desired color or click Other to create a custom color.
6	Click the down arrow  to the right of the Parameter Values entry field to open the color palette. Click the desired color or click Other to create a custom color.
7	Double-click the value in the CM Width entry field and type in the desired value.
8	Repeat Step 7 for the CM Height, SCM Width, and SCM Height entry fields to change the values, if desired.
9	Click the check box for the Load History Configuration function. • A **check** in the check box means the function is **enabled** and the configuration data on a block's Server History tab is loaded to the Server when the block is loaded to the execution environment. • A **blank** check box means the function is **disabled** and the configuration data on a block's Server History tab is not loaded to the Server.
10	Click the check box for the Load Trend Configuration function. • A **check** in the check box means the function is **enabled** and the Trend configuration data on a block's Server Display tab is loaded to the Server when the block is loaded to the execution environment. • A **blank** check box means the function is **disabled** and the Trend configuration data on a block's Server Display tab is not loaded to the Server.
11	Click the check box for the Load Group Configuration function. • A **check** in the check box means the function is **enabled** and the Group configuration data on a block's Server Display tab is loaded to the Server when the block is loaded to the execution environment. • A **blank** check box means the function is **disabled** and the Group configuration data on a block's Server Display tab is not loaded to the Server.

Step	Action
12	Click the check box for the Compare Parameters function. A **check** in the check box means the function is **enabled** and a **blank** check box means the function is **disabled**.
13	Click the check box for the Display Cross-References function. A **check** in the check box means the function is **enabled** and a **blank** check box means the function is **disabled**.
14	For the SCM Step Output Location function, • click the Left radio button to have the Step Output displayed on the left side of the SCM control chart drawing, • or the Right radio button to have it displayed on the right side of the SCM Control chart drawing.
15	Click the check box for the Enable Basic Version Control function. A **check** in the check box means the function is **enabled** and a **blank** check box means the function is **disabled** **REFERENCE - INTERNAL** If QVCS is licensed, then this option will not display. If QVCS is not licensed this option will be displayed.
16	Click the check box for the Use Conditional Propagation From Templates function. A **check** in the check box means the function is **enabled** and a **blank** check box means the function is **disabled**. When this function is enabled, changes to default parameters are propagated to sub-templates and instances. Default parameters are those parameters associated with a template that are not declared to be template-defining.
17	For the Relative Reference Display Option function, • click the Full Name radio button to have the relative reference displayed in the **full name** format (default setting), • or the Short Name button to have the relative refeence displayed in the **short name** format.

Related topics

Use Compare Parameters

Refer to Using Compare Parameters in this document.

Displaying Cross-References

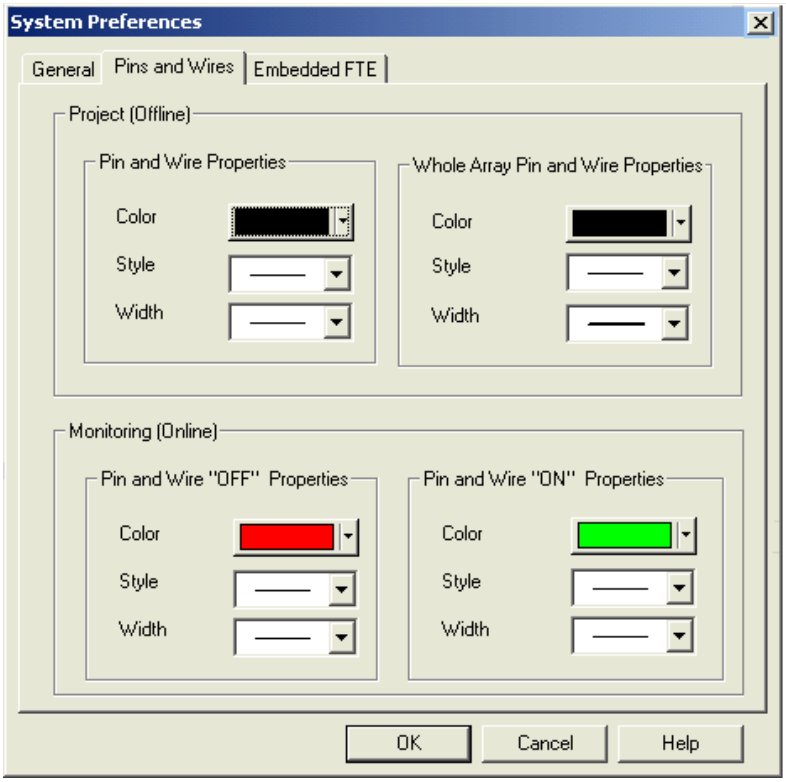
- Refer to Enabling cross-references in this document.

Relative Reference Display Option

- Refer to About Relative References

To establish Pins and Wires properties by using Systems Preferences dialog

Step	Action
1	Click the Pins and Wires tab to access it:.

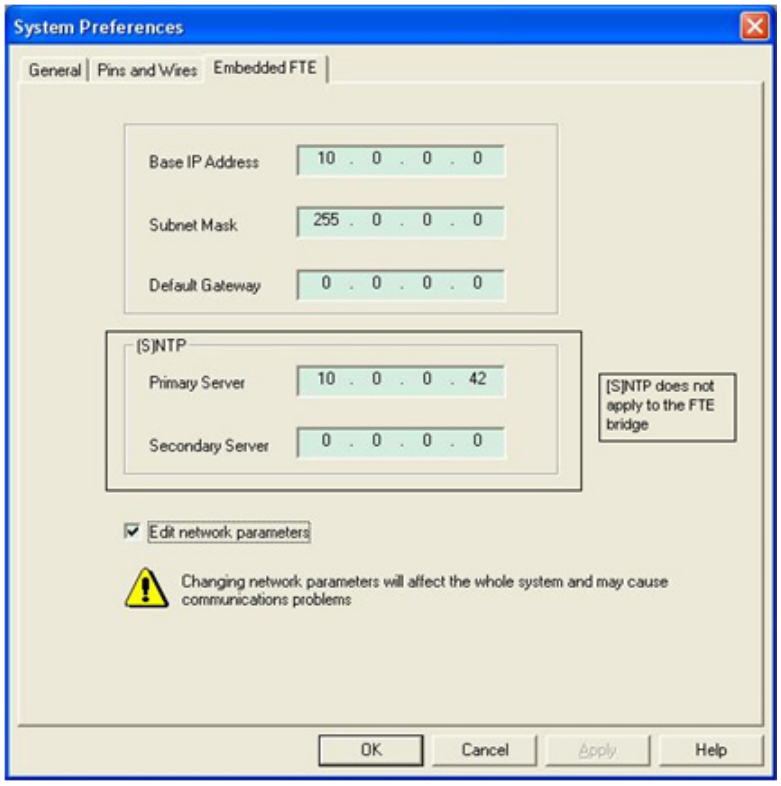
Step	Action
	
2	In Pin and Wire Properties category, click the down arrow to the right of the Color entry field to open the color palette. Click the desired color or click Other to create a custom color.
3	In Pin and Wire Properties category, click the down arrow to the right of the Style entry field to open the line style menu. Click the desired line style.
4	In Pin and Wire Properties category, click the down arrow to the right of the Width entry field to open the line width menu. Click the desired line width.
5	Repeat Steps 2 to 4 in the: • Whole Array Pin and Wire Properties,

Step	Action
	• Pin and Wire "OFF" Properties, and • Pin and Wire "ON" Properties categories to select the desired Color and line Style and Width.

To establish IP addresses using the Embedded FTE tab on System Preferences dialog

Step	Action
1	Click the Embedded FTE tab to access it.
2	Click the **Edit network parameters** check box to make address fields available for editing

Step	Action



TIP

If you have setup the FTE supervisory network, the Base IP Address should automatically reflect the correct **Network Number** and **Community Number** for the given FTE cluster. An IP address of 0.0.0.0 is invalid.

For address formats for **steps 3 - 8**, see Address configuration considerations

3 In **Base IP Address** box, type applicable address or accept the default. Press the TAB key.
Setting the Base IP Address prompts a suggested Subnet Mask address.

4 Type applicable **Subnet Mask** address or accept the default. Press the TAB key. An invalid Subnet Mask address displays an error message.

Step	Action
5	Type applicable **Default Gateway** address or accept the default. Press the TAB key.

ATTENTION

Steps 6 - 8 provide the procedure for establishing the **NTP time sources** for the C300 and the Series C FIM. The FTE Bridge Module does not use NTP, and the user is directed to **Step 9** if the network contains only FTE Bridge(s).

Selection of **NTP time sources** is discussed in

C300 and Series C FIM4 Time Synchronization.

6	Click the **Primary Server** box and type in the IP address for the primary time server.
7	If you want to specify a secondary time server, click the **Secondary Server** box and type in its IP address. Otherwise, you can skip this step.
8	If you specified a time server that is not on the local subnet, click the **Default Gateway** box and type in the IP address for the router port that defines the path to the server. Otherwise, skip this step.
9	Click the **Edit network parameters** check box. Clear (blank) the check box to make address fields unavailable for editing.
10	Click the **OK** button to save your changes and close the dialog.
11	Click the **OK** button to acknowledge the warning message about restarting remote Control Builder(s) after making changes in preferences.

ATTENTION

If you made address changes, be sure you stop and re-start the BOOTP Server service, so it does not use the old address as a reply to a module. Please refer to *Checking status of BOOTP server service* for information on how to access the **Experion BOOTP Server Properties** dialog to stop and start the service.

To change the Base IP Address updates IP information in ERDB

Typically, most users will not require their Base IP Address to be changed. But if changes to the Base IP Address are required, the IP information in the ERDB is now updated for the configured embedded nodes. Exchange block IP references and non-embedded FTE devices are not updated since they do not derive IP addresses from the system base IP.

Step	Action
1	Unload all devices from the Monitoring tab in Control Builder.
2	Change the base IP address in the System Preferences Embedded FTE tab.
3	Restart any remote Control Builder to refresh the view of the recently modified address configuration.
4	Reboot any embedded device, for devices that already received their IP addresses from previous established network configuration.
5	Load your system and continue.

Setting user preferences

Perform the following steps to customize the look of your Control Drawing by using the User Preferences dialog.

Prerequisites:

- Control Builder is running
- Tree windows are open

Considerations:

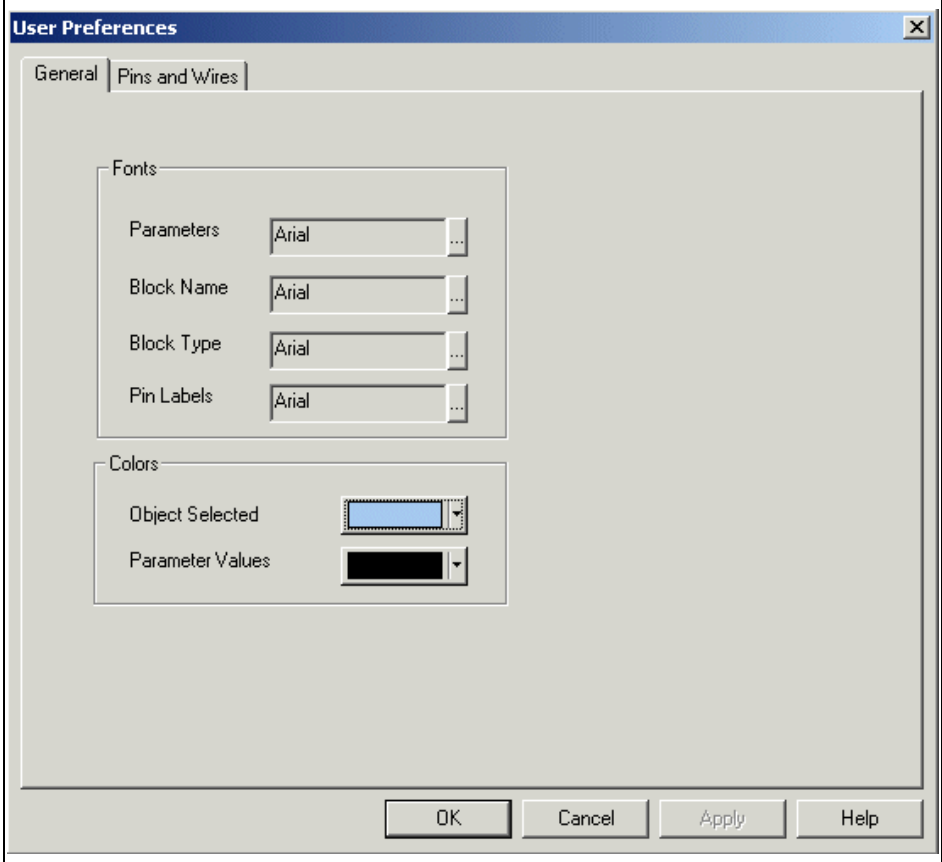
- All illustrations used in the procedure are for example purposes only.



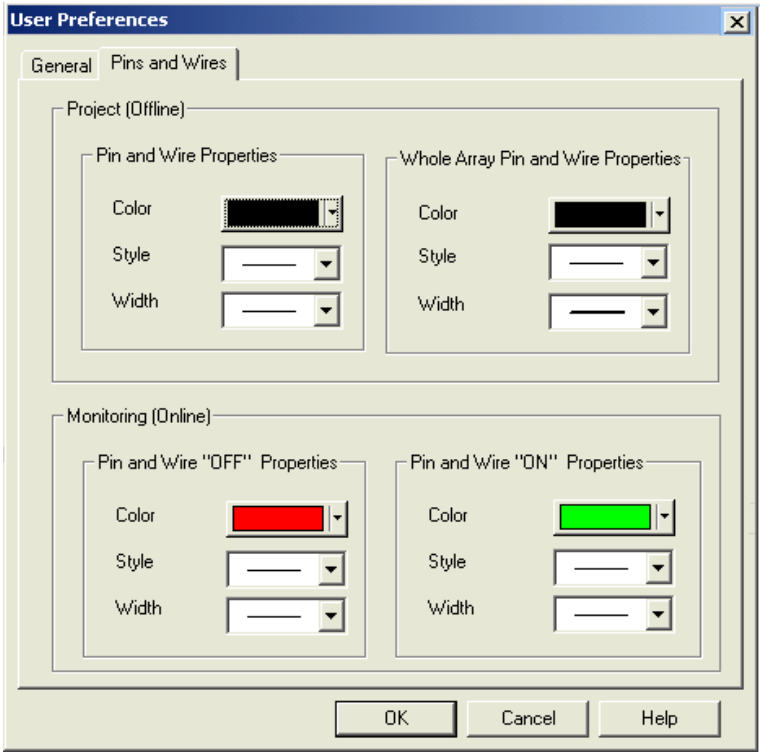
ATTENTION

- User preferences are stored in the registry and are intended for use by the current user.
- System preferences are stored in the database and are intended for use by the entire system.

Step	Action
1	Click Tools -> User Preferences to open the User Preferences dialog.



2	On the General Tab, click the Ellipsis button  to right of the Parameters entry field to call up the Font dialog box.
3	Select the desired font typeface, style, size and effects to be used for parameter data in Control Builder. Click OK to close the Font dialog.
4	Repeat Steps 2 and 3 to make desired Font selections for the Block Name, Block Type, and Pin Labels data in Control Builder.

Step	Action
5	Click the down arrow  to the right of the Object Selected entry field to open the color palette. Click the desired color or click Other to create a custom color.
6	Click the down arrow  to the right of the Parameter Values entry field to open the color palette. Click the desired color or click Other to create a custom color.
7	Click the Pins and Wires tab to access it: .
	
8	In Pin and Wire Properties category, click the down arrow  to the right of the Color entry field to open the color palette. Click the desired color or click

Step	Action
	Other to create a custom color.
9	In Pin and Wire Properties category, click the down arrow  to the right of the Style entry field to open the line style menu. Click the desired line style.
10	In Pin and Wire Properties category, click the down arrow  to the right of the Width entry field to open the line width menu. Click the desired line width.
11	Repeat Step 8 to 10 in the: • Whole Array Pin and Wire Properties • Pin and Wire "OFF" Properties, and • Pin and Wire "ON" Properties categories to select the desired Color and line Style and Width.
12	Click OK to accept the changes and return to Control Builder.

Control Builder Troubleshooting and Maintenance

Database maintenance

This section includes information that you can reference for:

Topic	Link
Deleting CM and SCM (optional)	Click here

Deleting CM and SCM (optional)

ATTENTION

The loaded CM exists in two databases. The loaded CM exists in the controller database that can be viewed from the Monitoring Tree of Control Builder. The loaded CM also exists in the “engineering database” that can be viewed from the Project Tree Window of Control Builder.

To truly delete a CM, the delete action must take place both from the Monitoring Tree View and from the Project Tree Window.

Follow the steps in the table below to delete a CM from the controller database.

Step	Action
1	Set up the Control Builder with Monitoring Tree Window visible and click the CM you wish to delete.
2	If the CM is Active, click Controller-> Inactivate -> Selected Item and then select Yes in the Change State dialog box to deactivate the CM.
3	Click Edit -> Delete. Alternate method: Press the <Delete> key.
4	Click Delete Selected Objects to delete the CM. The deleted CM will no longer appear in the Monitoring Tree Window.

Follow the steps in the table below to delete a CM from the “engineering database” from the Project Tree Window.

Step	Action
1	Set up the Control Builder with Project Tree Window visible and click the CM you wish to delete.
2	Click Edit -> Delete. Alternate method: Press the <Delete> key.
3	Click Delete Selected Objects to delete the CM. The deleted CM will no longer appear in the Project Tree View.

Appendix A - Smart Transmitter Integration with PM I/O

This section includes information that you can reference for:

Topic	Link
Introduction	Click here
Reviewing the Smart Transmitter Interface IOP	Click here
Integrating the Smart Transmitter	Click here
Reviewing DE communication protocol	Click here
What are IOP functions	Click here
Building point displays for ST devices	Click here
Using IOP cabling and FTA wiring	Click here
Field wiring the transmitter	Click here
Integrating multivariable field devices	Click here

Introduction

Honeywell Smartline Transmitters can be integrated into the Experion control system by employing Process Manager I/O components. The following sections describe the Smart Transmitter I/O Processor (IOP) and the installation considerations for integrating smart transmitters into the PM I/O subsystem.

Reviewing the Smart Transmitter Interface IOP

This section includes information that you can reference for:

Topic	Link
About digital integration	Click here
STI IOP features	Click here
Reranging	Click here
Bad database protection	Click here
Alarming	Click here
Off-line or on-line configuration	Click here
Transmitter diagnostics	Click here
Signal noise immunity	Click here

About digital integration

The Smart Transmitter Interface IOPs (STI IOP and STIMV IOP) provides digital integration of various Honeywell Smartline Transmitters into the Experion system. The STI IOP supports single-PV transmitters and the Smart Transmitter Interface Multivariable (STIMV).IOP supports both single-PV and multi-PV transmitters, (See Multivariable Field Device Integration for additional information).

A Smartline Transmitter can be set in analog or **Digital Enhanced (DE)** mode. DE mode offers bidirectional communication (read/write capability) between the system and the transmitter. In DE mode, it is possible to monitor and manipulate the transmitter from the operator station.

Examples of Honeywell Smartline transmitters are:

- ST 3000 Pressure transmitter
- STT 3000 Temperature transmitter
- MagneW 3000 Flowmeter
- SCM 3000 Smart Coriolis Flowmeter
- SMV 3000 Smart Multi-Variable transmitter

Each Smart Transmitter IOP (also referred to as the STI or STIM) can accommodate a maximum of 16 inputs from transmitters operating in the DE mode. The STI IOP accommodates sixteen single-PV transmitters. The STIMV IOP accommodates four multi-PV field devices with up to four PVs each, or some mix of single PV and multi-PV field devices that equals 16 inputs per IOP.

STI IOP features

The Smart Transmitter IOP provides the following features:

- Single window access to Smartline Transmitter data, including
  - Access to Smart Transmitter database using standard parameter access mechanisms (point and parameter),
  - Access to both Process Variable (PV) and Secondary Variable (SV) and PV conversion to user selected engineering units,
  - Support for reranging transmitters from the operator station or a control language program,
- Bad database protection to ensure the integrity of measurement data,
- Transmitter status access and alarm support,
- Range checking and PV filtering,
- Off-line configuration of Smartline Transmitters, including transmitter tag ID,
- Transmitter database upload, download, and checkpoint save,
- Display of transmitter self-diagnostic messages, serial and revision numbers, and the scratch pad,
- PV source selection—PV can be from the transmitter, a manually entered PV, or a program substituted PV, and
- Digital enhanced format makes the transmitter signal immune to noise.

By combining the Smart Transmitter IOP and Smart Transmitters operating in the Digital Enhanced (DE) mode, the transmitter operation and database become part of the Experion system.

These capabilities are aimed at improving the overall efficiency of the process plant operation by combining the central control room equipment with field equipment performance. A more complete list of digital integration features is outlined below.

Reranging

Reranged from the Smart Field Communicator (SFC), Smart Configuration Toollkit (SCT), or operator station, both the Upper Range Value (URV) and Lower Range Value (LRV) are accessible and modifiable.



ATTENTION

The Smart Field Communicator (SFC) is a hand-held, battery-powered unit and the Smart Configuration Toollkit (SCT) is a Windows-based PC application which are used for communication directly with the transmitter's database. Both the SFC and the SCT can be connected to the transmitter's output signal lines at any termination point,

Bad database protection

Bad Database Protection and Traffic Detection refers to the ability of the system to detect receipt of a bad database by comparing the database elements sent by the transmitter with the IOP memory database. If a mismatch of the database occurs, the transmitter database is flagged—and gives a bad PV signal to operator station to ensure that the control loop does not use a bad database.

The Smartline Transmitter broadcasts part of its database three times each second. The IOP compares the database received from the transmitter to the database stored in the IOP. If they do not match, the IOP flags the mismatched database and gives a bad PV signal to the operator station to ensure that the control loop does not use the bad database.

The Smart Transmitter IOP performs traffic detection — these are changes to the transmitter database from the hand held Smart Field Communicator (SFC). When the parameter being changed affects the PV, the IOP reports the bad PV to the operator station and the Experion System. This action prevents the control loop from operating on a bad database. It also alerts the system operator of the change to the transmitter database, which could impact the process operation.

Alarming

Digital integration of Smartline Transmitters gives full support of Experion process and system alarm mechanisms. The user can also configure alarm limits for Secondary Variables through additional point configuration. For example, you can configure a high temperature alarm for the ST 3000 pressure transmitter's Secondary Variable to detect any abnormal sensor temperature condition.

Off-line or on-line configuration

With Control Builder, transmitter configuration forms are available to build the transmitter database off-line. You can access the engineering personality off-line to construct the transmitter database—communication with the transmitter is not needed at this time. Refer to Creating PM I/O STIMV IOP for a procedure.

With the transmitter on-line, you can change transmitter parameters while monitoring a control strategy in Control Builder. Refer to Changing Parameters while Monitoring for a procedure.

Transmitter diagnostics

Smartline transmitters perform self-diagnostic functions. The results of the diagnostics are reported to either the Smart Field Communicator (SFC), when queried, or to the Station. The transmitter self-diagnostics information is displayed on the Point Detail display.

Signal noise immunity

A conventional 4-20 mA field instrument signal is subject to distortion caused by electrical noise or radio frequency interference. With digital communication, transmitter data is sent as a 24-bit floating point number. Noise immunity is achieved by filtering the noise from the signal by clearly separating signal levels—20 mA for logical zero and 4 mA for logical one. The transmission of data is determined to be either good or corrupted when a severe interference is present. This feature ensures the certainty of data used in a control loop.

Integrating the Smart Transmitter

The following is a complete checklist of tasks that must be accomplished to integrate the Smartline Transmitter and STI IOP into the Experion system.

Hardware considerations

The following hardware is required for Smartline Transmitter integration:

- STI or STIMV IOP—the Smart Transmitter Interface I/O Processor.
- FTA—the Field Termination Assembly connecting the field wiring to the IOP.
- FTA cable—the cable connecting the IOP and the FTA.
- Smartline 3000 Transmitter—a transmitter that can transmit in DE mode.
- Smart Meter (optional)—provides local display of the Smartline Transmitter output and loop diagnostics.
- Smart Field Communicator (optional)—hand-held device for local checkout of transmitter.
- Smart Configuration Toolkit (optional) -a Windows-based PC application for local checkout of the transmitter.

Bench check

The following is a list of tasks performed when bench checking, calibrating, and commissioning a Smartline 3000 Transmitter. The optional bench check procedure consists of the following:

- Unpack the transmitter.
- Check the contents.
- Connect a power source to the transmitter (with a 250 Ω resistor).
- Connect Smart Meter to the loop (optional).
- Run a communications check with the SFC or SCT.
- Check the operating status with the SFC or SCT.
- Check the configuration database with the SFC or SCT.
- Write to the scratch pad with the SFC or SCT.

- Check to be sure you have a transmitter with an adequate range for your application.

Commission loop

Communication check and commissioning the transmitter consist of the following:

- Test the communication loop to verify that the transmitter's configuration is consistent with the system's configuration.
- Check the operating status of the transmitter and loop.
- Check the output of the transmitter to be sure you are receiving the correct signal at its destination point.

Startup consists of the following:

- Calibrate zero on the transmitter to system conditions (except STT).

Reviewing DE communication protocol

This section includes information that you can reference for:

Topic	Link
DE versus analog format	Click here
Broadcast 4-byte format (PV)	Click here
Broadcast 6-byte format (PV, SV, and database)	Click here
Smart device communication architecture	Click here
Physical layer	Click here
Data link layer	Click here
Application layer	Click here
Floating point format	Click here
Error checking	Click here
Noise immunity	Click here

DE versus analog format

The Smartline Transmitters can be set in Digital Enhanced (DE) or analog communication format. Communications between an SFC, SCT or operator station and a transmitter communicating in analog format features a half-duplex, variable length message with a wake up pulse for on-demand requests and responses. While messages travel back and forth, the transmitter's output varies between 4 and 20 mA. For this reason, when in **analog format** your control loop must be in manual mode during SFC communications so the data exchange does not interfere with the control loop.



ATTENTION

In Digital Enhanced format, there is no need to put the control loop in manual mode when using SFC or SCT communication.

The Smartline Transmitters in DE format broadcast a new PV three times per second. When configured for Secondary Variable (SV) broadcast, the SV is transmitted every eighth broadcast, rather than the PV.

All communications between the Smart Transmitter IOP and the Smartline transmitters are in bit-serial form using the Honeywell DE (digital enhanced) protocol. The communication format for the transmitter is selectable through the **DECONF (DE configuration format).** The possible DECONF entries from the Control Builder are defined in the following table.

DECONF parameter

DECONF entry from Control Builder	Definition
Analog	Not supported by Smart Transmitter IOP
Pv	Transmitter communicates only the PV (4-byte format)
Pv_Sv	Transmitter communicates the PV and the Secondary Variable (SV) (4-byte format)
Pv_Db*	Transmitter communicates the PV and the database (6-byte format)
Pv_Sv_Db*	Transmitter communicates the PV, SV, and the database (6-byte format)

DECONF entry from Control Builder	Definition
* The use of these two formats is recommended for single PV and multivariable transmitters because they offer database mismatch detection and on-process mismatch recovery.	

Communication between an SFC, SCT, or a Smart Transmitter IOP and a transmitter communicating in the digital enhanced format features one of the message formats in the table above, depending on the operation being performed.

The DE message formats are selectable from the SFC and SCT using different designations: "single range," "dual range," and "single range with secondary variable." Select "Single Range with Secondary Variable" if the secondary variable is required. A second selection gives a choice of "w/DB (6- byte mode)" or w/oDB (4-byte mode)." The following table compares the DE message format and failsafe parameters as seen from the SFC and control builder.

Comparison of Control Builder and SFC/SCT DE configuration

If CB entry is . . .	Then comparable SFC/SCT entry is . . .	
DECONF	**Multivariable field device**	**Single PV transmitters(see notes)**
PV	Refer	Single or Dual Range, 4-Byte
PV_SV	to	Single Range W/SV, 4-Byte
PV_DB	Multivariable field	Single or Dual Range, 6-Byte
PV_SV_DB	device integration	Single Range W/SV, 6-Byte

NOTES:

1. 4-Byte - In R230, download is OK, but upload cannot be performed.
   In R300 and later, upload and download can be performed; the transmitter is automatically switched to 6-Byte mode.

   6-Byte - Upload and download can be performed.

2. FAILSAFE is not applicable from the Universal Station, but is user-configurable from the SFC:

   - upscale
   - downscale
   - freeze slot output

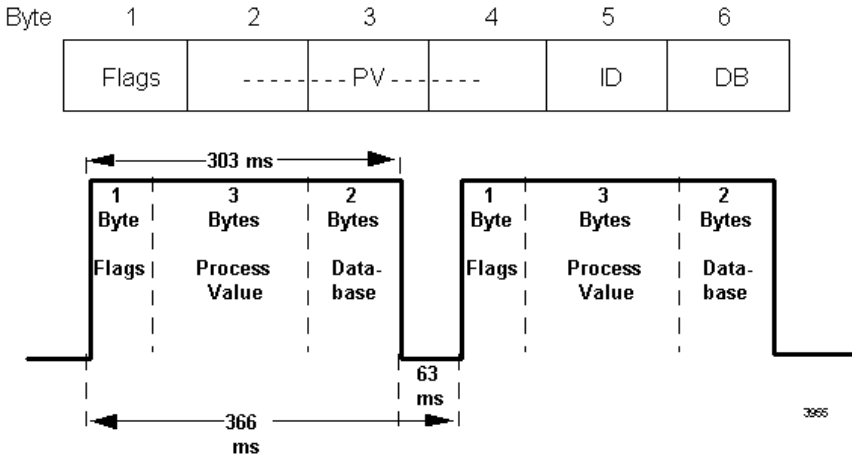
Broadcast 4-byte format (PV)

The 4-byte format is used for accessing the transmitter's process variable without database. It features one byte for flags, which includes transmitter status and some configuration data, and three bytes for process variable data. Flags include: Status, Failsafe Mode, 4/6 Byte Format, Variable Type (PV only, or PV and SV).

Broadcast 6-byte format (PV, SV, and database)

The following graphic illustrates the 6-byte message format. The 6-byte format is normally used for accessing the transmitter's PV, SV, and database. It is similar to the 4-byte mode format, but includes two additional bytes of transmitter database information to support both on-demand database upload requests from an SFC or SCT and passive database collection by the Smart Transmitter IOP. One of the two additional bytes is transmitter database so that a new database is received and

compared with the database in IOP memory every few seconds, depending on the transmitter type.



Typical timing for the 6-byte message format

TIP

For the Smart Transmitter interface, select the 6-byte format with Secondary Variable (SV) and database (PV_SV_DB), to take full advantage of the benefits of bidirectional digital integration

The following table lists the maximum broadcast database size and time for each transmitter type.

Flags include: Status, Failsafe Mode, 4/6 Byte Format, Variable Type (PV and Database, or PV, SV, and Database). The identification byte (fifth byte) contains the specific database location identifier. The database byte (sixth byte) contains one byte of database. SV is broadcast on every eighth broadcast in place of the PV.

Maximum transmitter broadcast database size and time

Transmitter Type	Database (bytes)	Time (Seconds)
ST 3000 (Pressure)	80	31
STT 3000 (Temperature)	92	35
MagneW 3000 (Flowmeter)	125	47
SMV 3000 Multivariable	up to 202	up to 74
SCM 3000 Multivariable Flowmeter	up to 255	up to 93

Smart device communication architecture

The Instrument Society of America Standard Practices Group # 50 established a standard for a digital, serial, bi-directional communication protocol for intelligent sensors and actuators mounted in an industrial process area. This standard defines a "four-layer" architecture: The Physical Layer, the Data Link Layer, the Application Layer, and the User Layer.

The Physical Layer defines the characteristics of the physical connection among nodes that are required to transport data from one device to another. Key elements of the physical layer include: topology of the network and its limitations, modulation techniques, data rates, redundancy requirements, and physical media types.

The Data Link Layer defines the media access control and data movement and scheduling function. This layer, also addresses mechanisms for error detection and retry policy.

The Application Layer defines the user interface message formats and services available with this communication protocol.

The User Layer defines the application oriented database, such as analog input and control functions.

The layers of the Honeywell DE Protocol are defined in the following paragraphs.

Physical layer

The **Physical Layer of DE Protocol** defines the mechanical and electrical characteristics of the DE Protocol physical interface. The physical connection between the Smart Transmitter IOP FTA and the Smartline Transmitter is through a twisted pair wire. The field devices are typically powered from 24 Vdc.

Information is exchanged by **NRZ (Non-Return-to-Zero) modulation** of the loop current. Changes in loop current can be sensed at the field device or across a 250 ohm (load) resistor placed in series with the loop and physically located between the field device and supply return. Note that the signal will be inverted when sensed across any series device. Active communicating host devices (SFC, SCT or operator station) must be electrically connected across the 4-20 mA wires feeding the field device (transmitter), with a minimum of 250 ohms resistance between it and the loop power supply.

The resistance inserted into the loop should be a minimum of 250 ohms. Larger resistance values can be used, provided they still meet the overall system requirements.



WARNING

Make sure you do not power starve the field device. A steady and reliable power source is recommended for all field instruments.

Signaling levels

With the introduction of the DE Protocol, the traditional analog signal used to communicate the PVs from the transmitters to the controllers has been greatly altered. The continuous 4-20 mA analog PV has been replaced by a continuous digital PV signal.

logic 0 = 20 mA (5 V, typically across 250 ohm load resistor)

logic 1 = 4 mA (1 V, typically across 250 ohm load resistor)

The communication signal's rise and fall times are intentionally slow in order to reduce potential crosstalk problems within large wire bundles.

$T_{(rise)}$ = ~0.5 ms, ±20% (0-63.2%)

$T_{(fall)}$ = ~0.5 ms, ±20% (0-63.2%)

Performance considerations

The signaling and data rate is 218.5 bps, ±1%. The ±1% tolerance is the sum of all error contributors on a communication link (slave and host, for example). It is typically assumed that the field transmitters contain lower cost/performance resonators with worst case tolerances of ±.99% and that the host devices contain higher cost/performance crystals with worst case tolerances of ±0.005%. It also takes into consideration, performance with long cable lengths and a worst case timing jitter of 1/4 bit.

Data link layer

The **Data Link Layer of DE Protocol** defines the data flow initialization, control, termination, and error recovery. The Data Link Layer protocol is half-duplex, bit-serial, byte oriented. For every request from a master, there must be a response from the slave. Allowances for up to three retries are incorporated for harsh or noisy environments.

The following gross status responses result in a retry from the transmitter:

- Nack
- Busy
- Illegal response

The following communication error/faults result in a retry:

- Bad checksum
- Bad parity
- Incomplete message
- Framing error
- No transmitter response

Application layer

The **Application Layer of DE Protocol** functions include:

- Detailed Transmitter Status
- Digital PV/Transmitter Status
- Bad PV Protection
- Bad Database Protection
- Long PV Dropout
- Database Download
- Database Upload
- Initial Database Check — Power up
- Database Mismatch

- SFC Traffic Detect

Floating point format

The Honeywell Floating Point Format is 24 bits (3 bytes) in length for PV and SV representation and includes the following:

• One Sign bit (0 = positive, 1= negative)

• Seven bits of Exponent (excess 64)

• Sixteen bits of Mantissa

• Fixed implied binary point

The number is stored in three bytes as shown below.

Sign	Exponent	Mantissa
1 bit	7 bits	16 bits

The resolution of a number (worst case) is 1/32768 or 0.003% of any number.

Error checking

The DE protocol incorporates two means of enhancing error detection to ensure that the original information is correctly received by the system:

• **Field Device Status**—The DE protocol requires that an indication of the field device status be transmitted with every digital PV value. As a result, the system will never control on a PV from a known bad field device.

• **Information Redundancy**—Redundancy is the addition of noninformation carrying bits so that calculations can be made at the receive end to determine whether the group of bits, including the redundant bits, are the same as those transmitted.

Noise immunity

In DE protocol, increased noise immunity is gained by using a digital communication baud rate that is low in frequency relative to its environmental noise. The DE protocol transmits at a rate of 218 baud. The advantages of this are

1. High frequency noise can be easily filtered out with simple RC filters (on the Smart Transmitter IOP board).

2. It is less likely that burst noise will last long enough or contain enough energy to interfere with the signal.

Noise immunity is also gained by selecting significantly large and different voltage/current levels to represent a logical 1 and 0. For the DE protocol, these levels are 4 mA/1 volt and 20 mA/5 volts.

What are IOP functions

The primary function of the **Smart Transmitter Interface I/O Processor (STI and STIMV IOPs)** is to provide up to sixteen independent bidirectional channels of communication between the operator station and the Smartline Transmitters in DE mode.

The Smart Transmitter IOP functions include:

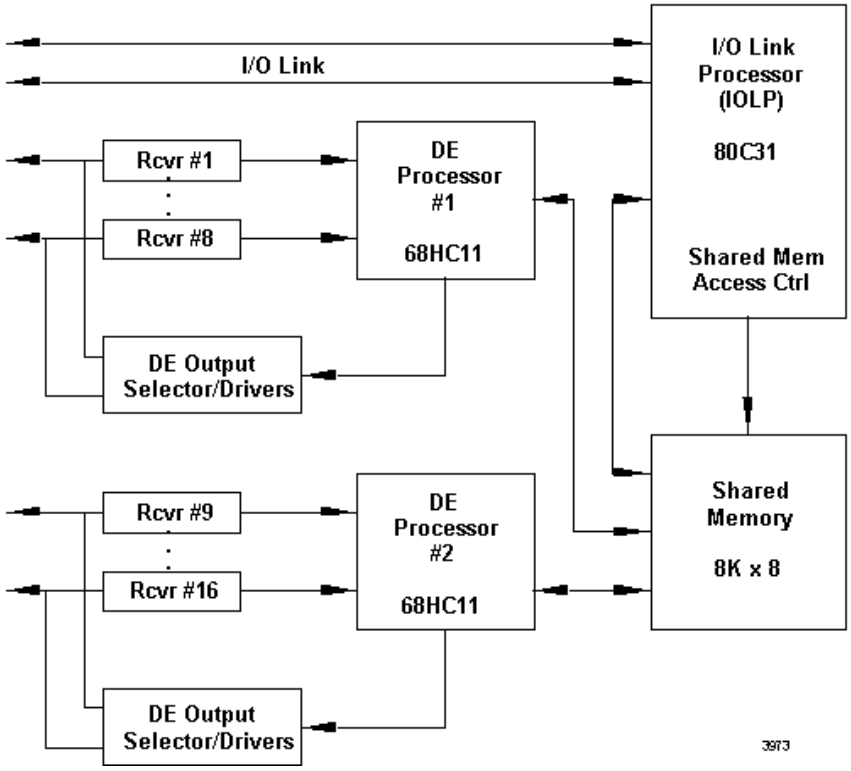
- Ground-referenced transmitter power for each input channel.
- Filtering input signals for high frequency noise suppression and surge limiting.
- Slew rate limiting for outputs to Smartline transmitters to minimize cross talk and interference with other system signals.
- Receiving and decoding serial input data on each channel.
- Sending configuration commands to Smartline transmitters.
- Converting digital input representation into Engineering Units (EUs).
- Making processed input data available to devices on the I/O Link in the form of EUs and/or percent of range.
- Maintaining local database of configuration and processing information.
- Checking received data against specified limits, maintaining alarm flags, and generating alarm events.
- Providing an orderly, initialized state of limited operation, and accepting downline loaded configuration data from higher-order devices on the I/O Link to establish full functionality.
- Interface to the I/O Link, with full support of the I/O Link protocol.
- PV Source selection.
- Optional IOP redundancy
- Power-on database validity detection

The **two DE Processors** on the IOP each provide communication with eight DE Smartline transmitters. Each DE processor consists of

- A processing unit, comprised of a Motorola MC68HC11 single chip microcomputer, 32 K bytes of ROM, and 8 K of RAM.
- Eight data receivers, which provide input signal conditioning (noise filtering, surge limiting, and so on) for the serial data inputs on the processing unit.
- An output channel selector and driver, which allows the DE processor to talk (output) to any one of its eight connected devices.

The IOP's 8 K byte Shared Memory provides a mailbox through which the I/O Link Processor and the two DE Processors can communicate. Access to the shared memory is under the control of the I/O Link Processor.

The IOP's Field Termination Assembly (FTA) includes 250-ohm range resistors to convert the 4-20 mA current signals from Smartline Transmitters in DE mode, and the 4-20 mA output signals from the IOP itself, into 1-5 volt signaling levels.



Smart transmitter IOP functional block diagram

Input signals are stripped of high frequency noise by single pole 12 kHz RC filters on the IOP.

DE Processor functions include:

- Error detection
  - 5 time bit sample
  - 4 of 5 sample voting
  - Parity (odd)
  - Framing (1 stop req. >1.4 stop bad)
  - Data context checks
  - Checksum on database
- Database formatting
  - floating point to IEEE
  - Packed BCD to ASCII
- Two copies of PV, SV, and database for each slot
  - Last complete valid copy
  - Active copy (currently being received)
- Pass valid PV, SV, and database to I/O link processor via shared RAM

IOP and transmitter database

The transmitter database can be configured at the operator station and downloaded to the transmitter, or the transmitter database can be uploaded to the IOP, as required, when the Smart Transmitter point is in the inactive state. During normal operation (when the Smart Transmitter point is in the active state), each time that the transmitter broadcasts the PV value to the IOP, it also sends the one byte of its database (depending on the selected DECONF mode) to the IOP. This allows the IOP to compare the stored database to the newly received database to check for database discrepancies (mismatch). If a discrepancy is detected, the PV is set to NaN and the status is set to DBCHANGE. The user can easily correct the discrepancy by uploading or downloading the database.

All key transmitter parameters can be accessed from the operator station including

- Upper and Lower Range Values
- Damping
- PV type
- DE (Digital Enhanced communications) configuration variables
- Status of the Transmitter
- Transmitter's serial number and software revision number
- Transmitter's scratch pad

The user can access these variables through the point's Detail Display.

The Smart Transmitter IOP maintains a copy of the transmitter's database. When a transmitter failure occurs, the database can be downloaded to the transmitter. This database download feature can significantly reduce the downtime of a control loop by reducing the time to get a replacement transmitter into operation.

The Smart Transmitter IOP also allows the user to access the detailed status of a transmitter. The operator station displays the transmitter status and the scratch pad information that has been entered, including any maintenance notes. This remote diagnostic feature greatly improves the safety of maintenance personnel, by reducing the need to physically access transmitters located in hazardous areas.

Calibration of the transmitter can also be accomplished from the operator station. This function allows on-line adjustment of the transmitter's working ranges so that the reference points for a measurement are accurate.

In addition, a Smart Field Communicator (SFC), which is a hand-held device, can also be physically connected to the appropriate FTA in the control cabinet to communicate with Smartline Transmitters without disrupting the process, as required (refer to the *Operating Guide for Smart Field Communicator*s).

Building point displays for ST devices

See Creating PM I/O STIMV IOP for details on building point displays for smart transmitters.

Using IOP cabling and FTA wiring

The Smart Transmitter is connected to a Field Termination Assembly (FTA) in the control cabinet through twisted pair wires on a point-to-point basis. The FTA is cabled to the Smart Transmitter Interface I/O Processor board (either STI or STIMV IOP).

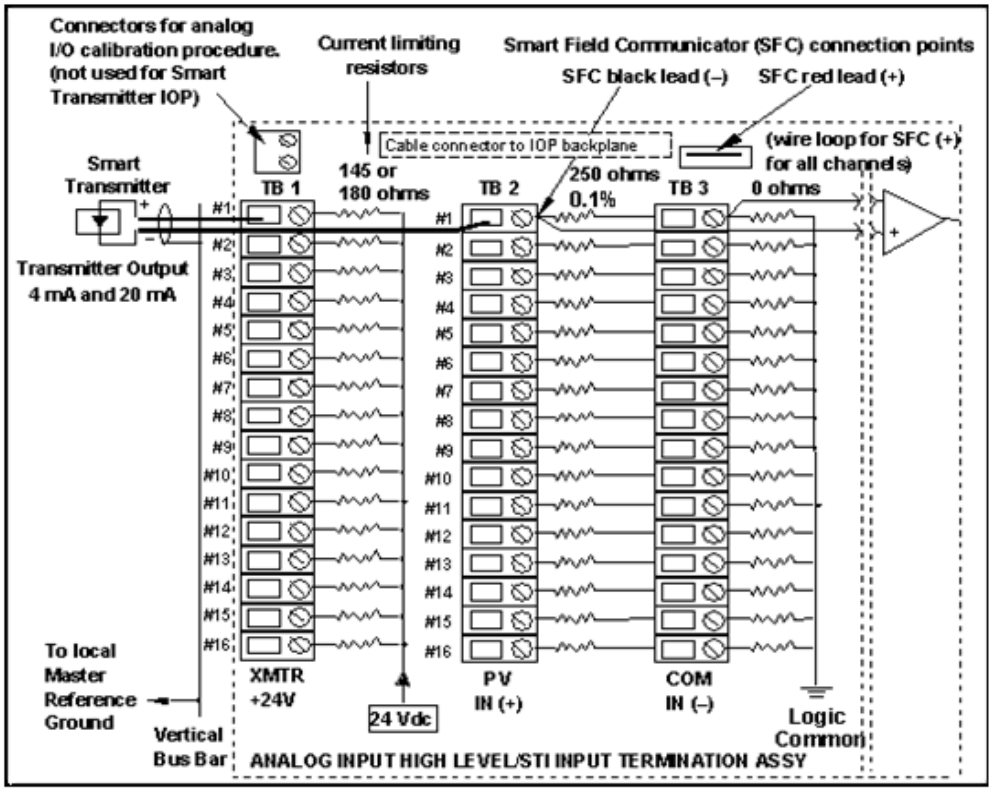
Selection of cable lengths should follow Honeywell's guidelines. The FTA cable is not to exceed 50 meters. See the following sections on redundant, intrinsically safe, and nonredundant Smart Transmitter IOPs, for Honeywell recommended cabling.

FTA wiring

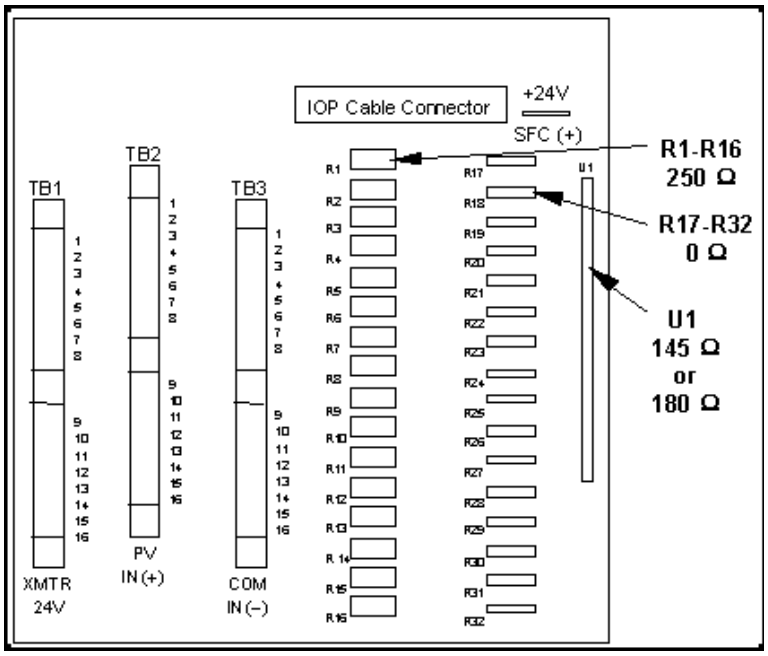
The field termination assembly (FTA) for the smart transmitter is the same as that used for high level analog inputs. There are two types of FTA: one supports redundant IOPs, and the other does not. The redundant and the nonredundant versions of the FTA can use screw-type or Weidmuller compression type termination.

Figures 3 through 6 illustrate the FTA and circuit wiring for the nonredundant Smart Transmitter IOP. Figures 8 and 9 illustrate circuits for the redundant Smart Transmitter IOP.

Notice the SFC connection points on the FTA in the following graphic.



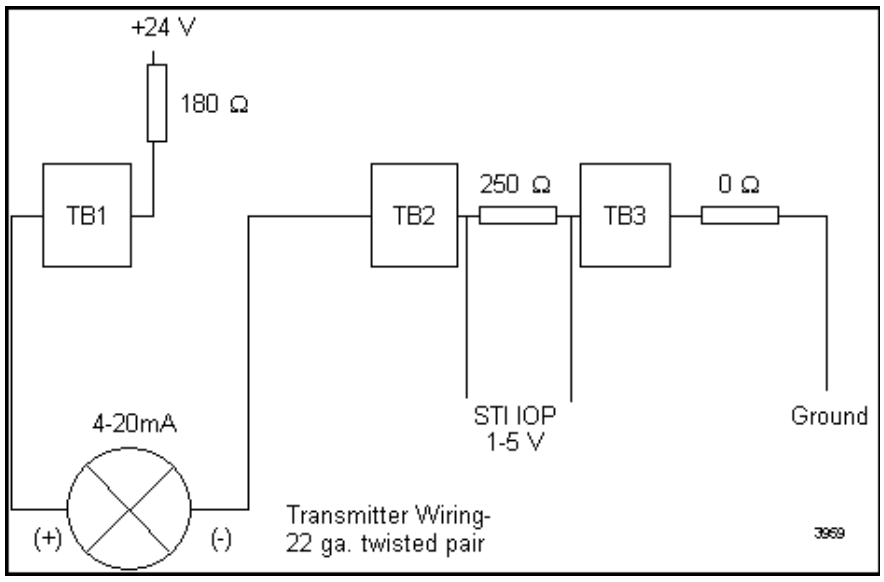
FTA Model MU-TAH02 schematic - nonredundant IOP



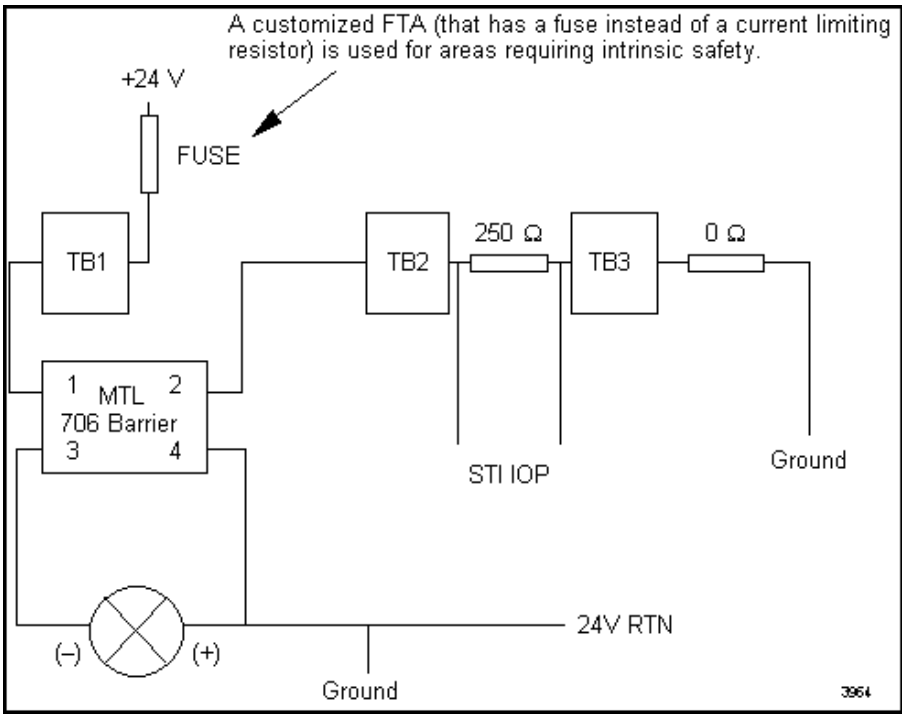
FTA Model MU-TAH02 physical layout - nonredundant IOP

ATTENTION

For schematic diagrams and assembly layouts of other applicable FTAS, see AI Wiring in Appendix C and HLAI GI Wiring in Appendix D of the Control Hardware Installation Guide.



Nonredundant IOP circuit for smart transmitter (Non-I.S. environment)



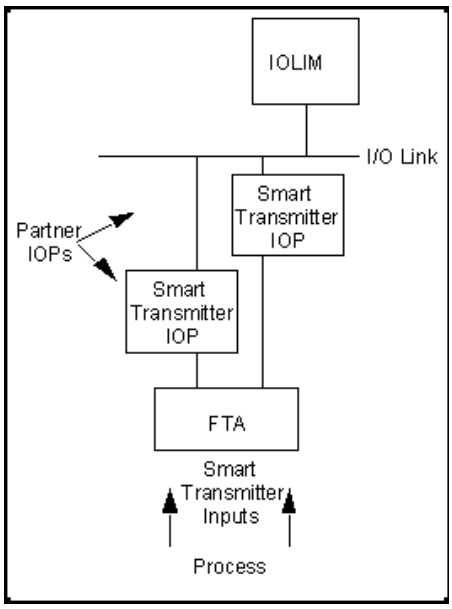
Nonredundant IOP circuit for smart transmitter (I.S. environment)

Redundant IOPs

The Smart Transmitter IOPs are available as redundant partners. Either one of the partner IOPs operates as the primary and the other backs up the primary. Each of the partners connects to the process through a single Field Termination Assembly (FTA).

Both the primary and backup IOPs receive all data from the operator station or from the process simultaneously, and should the primary IOP fail, the backup IOP takes over automatically, becoming the new primary IOP. Such a failover is transparent to the remainder of the system, except for operator station displays that show IOP status information. A failover is completed in 100 milliseconds or less.

At the operator station Detail Status display, you can request that the primary and backup roles for the partner IOPs be exchanged. To do this, select the IOP, then select `RUN STATE`S, `SWAP PRIMAR`Y, and `ENTE`R.

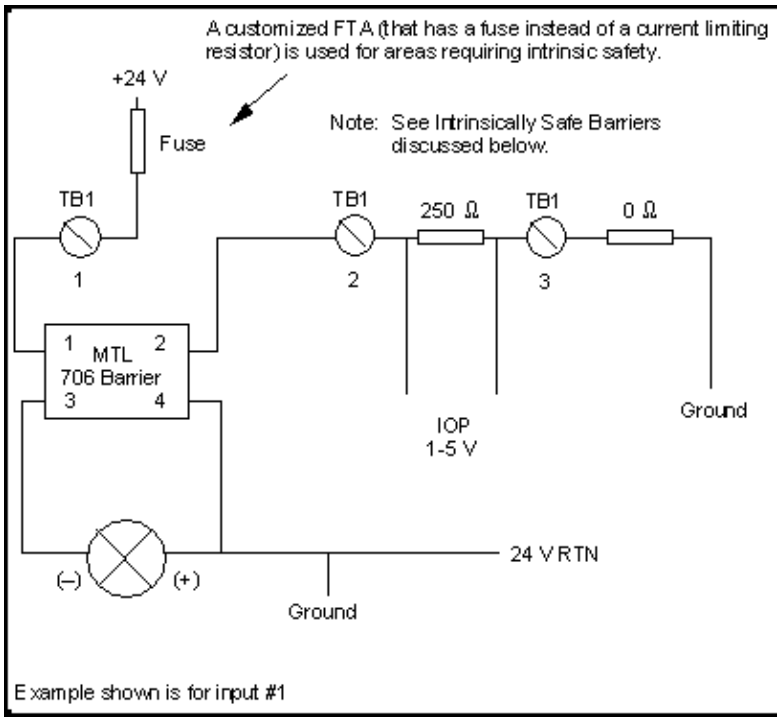


Redundant smart transmitter IOP signal flow

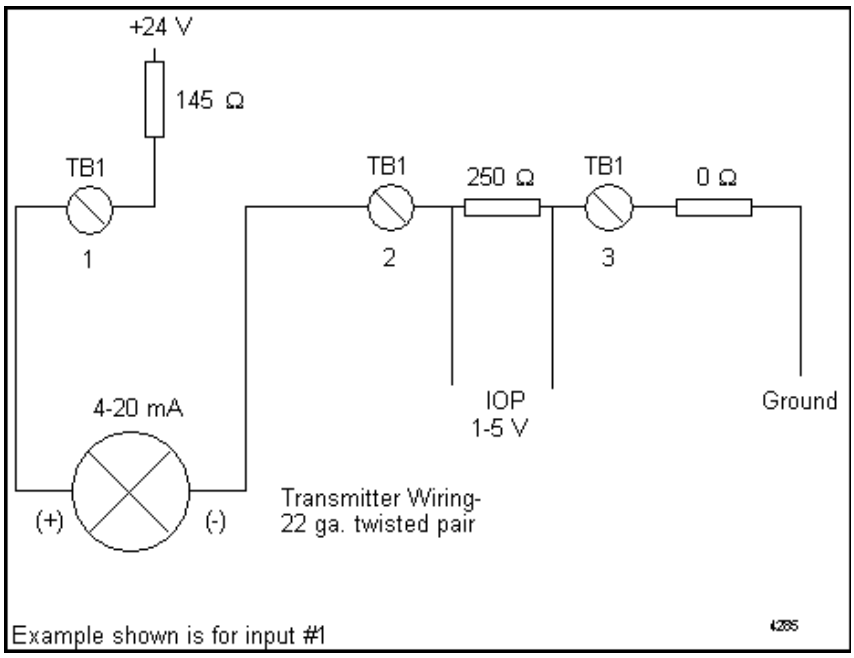
Synchronization of the database in the partner IOPs is verified as the backup IOP checks that the primary IOP received and responded to each data item sent on the I/O Link, and by periodic comparisons of the databases in both IOPs.

Smart Transmitter IOPs monitor the inputs from the FTA. A diagnostic routine verifies that the physical interconnection between the partner IOPs is functional.

In a redundant configuration where a STIMV IOP is used, both IOPs must be STIMV IOPs. See Upgrading STI IOP to STIMV IOP for more information.



Redundant smart transmitter IOP circuit (I.S.) - screw type FTA



Redundant smart transmitter IOP circuit (Non-I.S.) - screw type FTA

Intrinsically safe barriers

Intrinsic safety barriers can be implemented using traditional Zener barriers or using Galvanic Isolation/Intrinsic Safety Field Termination Assemblies (GI/IS FTAs) that are available with PM IO.

On the GI/IS FTAs, plug-in modules, called isolators, incorporate galvanic isolation and intrinsic safety functions. The GI/IS FTAs are always fully populated with isolators to provide a one-to-one details for the GI/IS FTAs (model numbers: GAIH13-compression type and GAIH83-crimp type). The GI/IS FTA Specification and Technical Data provides additional information correspondence with the respective IOP. The Smart Transmitter GI/IS FTA has 16 isolators. Appendix D in the Control Hardware Installation Guide provides complete drawings and wiring (document number GA03-100).

The following paragraphs discuss traditional Zener barriers.

Honeywell has tested five intrinsically safe barriers to handle bidirectional (two-way) communications for Smartline Transmitters interfaced with a Smart Transmitter IOP to provide a few safe operations.

The current mirror barriers approved by Honeywell include:

- MTL 706 and 3046
- Elcon CS-I-7042
- Pepperl & Fuchs KHD3-IST/EX 1 or KHD3-ISV/EX 1
- Stahl 9001/51-280-091-00

The galvanic isolation barrier approved by Honeywell, is not as wide ranging as the current mirror barriers and is used primarily in Europe:

- Stahl 9603/712213 C854

It is the user's responsibility to verify that any part used has the appropriate safety rating. When installing Honeywell suggested I.S. barriers with bidirectional mode you must make the following power accommodations:

- Barrier leads must get 24 volts directly. There must be no resistance in the 24 volt lead.
- Barriers themselves draw power. You may have to allocate as much as 100 mA per transmitter to compensate for this power draw.



ATTENTION

For Division 1 applications requiring the use of Intrinsic Safety Barriers, a special FTA (without a 180 Ω current limiting resistor) is required. Also, the particular model of barrier must be selected carefully so that it will not filter out the digital communication. Consult your Honeywell representative for further information and recommended barrier models. Refer to Figures 5 and 7 for the MTL 706 Barrier.

MTL 706 and 3046 IS - The MTL 706 barrier is recommended for use with Smart Transmitter IOP. The barrier passes communication bidirectionally and works properly with the SFC or DE meter in any location. It provides a regulated voltage of 23.6 Vdc out to the field for any power supply voltage between 22.3 Vdc and 35 Vdc. The barrier will handle at least 200 ohms of line resistance. A sample barrier demonstrated a maximum dc current mirror error of 0.0007%. Like most barriers, the bias current required to operate the barrier is roughly twice the signal current.

The MTL 3046 IS barrier is also acceptable. It will pass communications bidirectionally and has plenty of voltage and resistance margins to handle most applications.

Elcon CS-I-7022/IIS/ST - This barrier operates properly with any loop resistance between 0 to 170 ohms. The power supply considerations is not an issue with this barrier because it operates from an 120 Vac line. It generates internally a 24 Vdc transmitter supply.

Pepperl & Fuchs KHD3-IST/EX or KHD3-ISV/EX 1- The two barriers are twins. The ISV version has a voltage (as opposed to a current) output; otherwise, they are identical. Both barriers were tested over a power supply range of 18.5 volts to 35 volts and loop resistance of 0 to 220 ohms. They both performed acceptably under all conditions.

Stahl 9001/51-280-091-00 - This current mirror barrier is an acceptable barrier. Check the manufacturers catalog to verify the exact model number for your application.

Stahl 9603/712213 C854 - The Stahl 9603/712213 C854 galvanic isolation barrier is acceptable for service with Smartline Transmitters, the Smart Transmitter IOP, and the IOLIM, with certain limitations. These limitations include the following:

- The Smartline Transmitters cannot be located more than 1,000 to 1,500 feet of wiring from the I.S. barrier. The voltage available to the field device is only 12.4 volts during communications. This limits the resistance of field wiring to 70 ohms for the ST 3000 Pressure Transmitter and 20 ohms for the STT 3000 Temperature Transmitter. These are quite low values, but 20 ohms is equivalent to 1,000 double feet of 20 AWG or 1,500 double feet of 18 AWG.
- Since the barrier can draw over 200 mA, it cannot be powered from the FTA. The Stahl I.S. Barrier card file must be powered directly from the 24 volt power supply. The 24 V power to the card file requires separate fusing. In addition, the amount of power required for the barriers must be considered in the overall power budget for the system.
- Using the SFC on the safe (control room) side of the barrier will require that the power return and the signal ground be tied together somewhere in the system. This is required because the SFC signals with a current and needs a current return path. There is no such limitation when using the SFC on the hazardous side of the barrier.

Check the manufacturer's catalog to verify the exact model number for your application.

Field wiring the transmitter

For field wiring of the smart transmitter, refer to the user manual for that specific field device.

Integrating multivariable field devices

This section includes information that you can reference for:

Topic	Link
STIMV IOP	Click here
FTA wiring rules	Click here
Configuration rules	Click here
DECONF parameter	Click here
Field device in analog or 4-byte mode	Click here
DECONF configuration rules	Click here
Download	Click here
Database discrepancies	Click here
Upgrading STI IOP to STIMV IOP	Click here
Synch error	Click here

STIMV IOP

Multivariable (multiple PV) smart field devices are supported using the Smart Transmitter Interface Multivariable (STIMV) IOP. The STIMV IOP supports all Honeywell Smartline field devices, but certain precautions must be followed when integrating multivariable field devices.

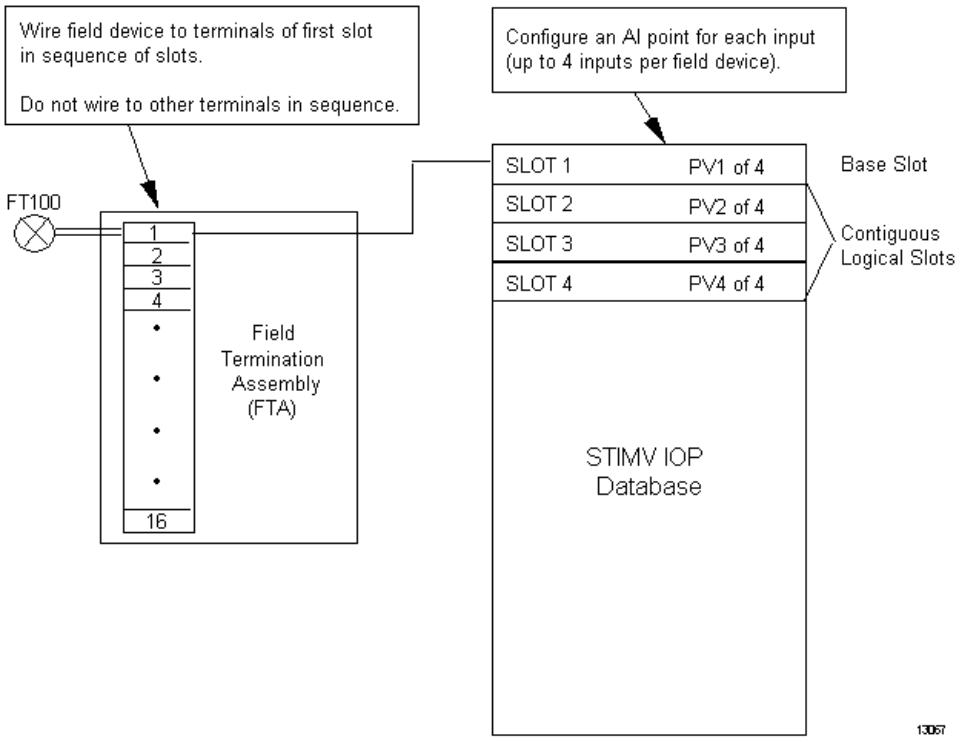
FTA wiring rules

Multivariable field devices are wired for integration the same way you would any other Smartline transmitter.

If the field device will provide multiple inputs (PVs), the FTA screw terminals used for the transmitter's DE output connection identify the physical slot for the transmitter's PVs. In this case, be sure

- No other Smartline transmitters are connected to contiguous FTA screw terminals that are allotted as logical slots for the transmitter's other PVs.
- Contiguous logical slots allotted for the transmitter's other PVs do not cross over IOP boundaries from 8 to 9 or wrap around an IOP boundary from 8 to 1 or 16 to 9

For example, a multivariable field device physically connects to one STIMV slot (the base slot). The user allocates up to three additional contiguous slots, depending on the number of inputs (PVs) desired from the field device. The figure below illustrates the configuration of a 4-input multivariable field device.



Multivariable field device slot requirements

ATTENTION

For schematic diagrams and assembly layouts of other applicable FTAS, see AI Wiring in C and HLAI GI Wiring in Appendix D of the Control Hardware Installation Guide.

Configuration rules

Follow these rules when integrating multivariable field devices:

- No field device should be physically wired to the FTA screw terminals belonging to any slot other than the first slot in a logical sequence of slots.
- When assigning contiguous slots for a field device, you must adhere to the STIMV IOP's DE processor slot boundaries, that is, the contiguous logical slots must fall in the range of 1-8 or 9-16. *The contiguous slots cannot overlap boundaries (from slot 8 to 9) or wrap around (from 8 to 1 or 16 to 9).*
- The STITAG parameter must be identical and nonblank across all logical slots configured for use by a multivariable field device. If the STITAG parameter is null or all blanks, the IOP assumes it is a single variable field device.
- The STIMV IOP determines the number of PVs being sent by a transmitter based on the number of contiguous, identical STITAG parameters.
- If the STIMV IOP detects contiguous slots with the same STITAG parameters, it recognizes the first slot in the sequence as the base slot for the field device. On page 2 of the Point Detail display, the base slot is referred to as PV 1 of *n*, where *n* is the total number of contiguous slots with identical STITAG parameters. The base slot should have the hardwired connection to the transmitter. The remaining slots allocated should not have a hardwired connection.

If you use the SFC or SCT to configure the number of PVs to be sent by the field device, the configuration must match the number of slots allocated in the IOP (slots with identical STITAG parameters).

- FTA communications are through the base slot, however, only the slot being implemented needs to be set active. For example, if you want the percent solids for the SCM 3000, you must allocate three slots, but only the slot for PV 4 needs to be active.
- After adding or deleting logical slots, you must download from the base slot that is physically connected to the field device in order to change the number of PVs being transmitted (DECONF).
- In order to download configuration of any slot in a multivariable field device, you must make all of the associated slots INACTIVE.
- Upload of a multivariable transmitter that is in analog or 4-byte mode must be performed from PV 1.

DECONF parameter

The DECONF parameter of the base slot and each contiguous logical slot for a multivariable transmitter must agree as defined in the following table. If they do not agree, a DECONF database discrepancy results. Note that if using an SFC or SCT, the comparable entries listed in table below must also agree for the base slot and each logical slot of a multivariable transmitter.

DECONF Parameter Entry Rules

If PV1 is . . .	The PV*n* must be . . .
PV	PV
PV_SV	PV
PV_DB	PV_DB
PV_SV_DB	PV_DB
where n = 2, 3, or 4	

ATTENTION

SV is available only for PV1.

Field device in analog or 4-byte mode

Attempting to upload a multivariable transmitter that is in analog or 4-byte mode from a slot other than PV 1, causes a Transmitter Status error message to appear:

```
COMMAND ALLOWED ONLY ON FIRST
SLOT OF MULTIPLE PV XMTRS
```

The error message appears because the STIMV point, before performing the upload, attempts to download the DECONF configuration to establish the 6-byte digital mode, but DECONF changes can only be made from PV 1.

In response to the above error message, the user should do the following:

Step	Action
1	Perform an upload or download from PV1, changing DECONF to 6-byte digital mode.
2	Perform an upload from PVn.

DECONF configuration rules

The following rules apply to the DECONF parameter for multivariable transmitters

- Perform upload or download of DECONF from the base slot of a multislot configuration (PV 1 of *n*).
- Download a change in the DECONF parameter of any slot in a multislot configuration from the Detail display of PV 1 of *n*.

A DECONF parameter change to turn PV 2, 3, or 4 ON (which is equivalent to building a point for the given transmitter PV) can only be downloaded from PV 1. If you try to download a DECONF change from the Detail display of PV 2, 3, or 4, you will get an error message

If a transmitter sends PV data that overlaps another transmitter's data, the following error indications appear

- DECONF appears in the DATABASE DISCREPANCY area of the Detail display, indicating a database mismatch,
- DBCHANGE appears as the transmitter STATE on the Detail display,
- this Transmitter Status message appears on the Detail display:

```
CONFIGURATION MISMATCH
MULTIPLE DEVICES ASSIGNED TO SLOT
```

- one or two asterisks appear in the `x of n` display for all logical slots associated with the conflicting devices.

Operator intervention is required to clear the discrepancy by disconnecting the offending device or reconfiguring the number of PVs to eliminate the slot overlap (see Configuration rules for how to configure DECONF to specify the number of PVs).

PVCHAR parameter

PV characterization is assumed to be Linear for all multivariable field device PVs associated with inputs without recognized base units.

PVEUHI/PVEULO parameter

Use the mx+b equation to convert the PV to engineering units other than the base engineering units.

SENSRTYP parameter

For each slot of a multivariable device, select the SENSRTYP specified in the user manual for the transmitter.

Note that many slots of multivariable transmitters have a real SENSRTYP other than the configuration choices (SPT_DP, SPT_AP, SPT_GP, STT or SFM). For these slots, choose SFM for the SENSRTYP. When the transmitter is connected, the TRANSMITTER STATUS field on the point detail display will indicate the actual SENSRTYP.

STI EU parameter

For multivariable transmitter slots with SENSRTYP of SPT_DP, SPT_AP, SPT_GP, or STT, choose the preferred STI_EU (engineering units). For slots with SENSRTYP of SFM, select BLANK or CM/HR if BLANK is not available. When BLANK (or CM/HR) is selected, the limit values URL, LRL, URV, and LRV are always displayed in the base engineering units specified in the transmitter user manual.

The 8-character EUDESC parameter can be used to indicate the correct units of the PV to the operator.



ATTENTION

All points of a multipoint transmitter should be made INACTIVE when altering the database of any or all points in the transmitter.

Download

Each logical slot associated with a field device controls its own configuration parameters. On a per-slot basis, the user can change parameters in the transmitter database and download them from the slot's Detail display (DNLOADDB command) with the exception of DECONF.

Upload and download of DECONF can only be performed from the base slot (PV1 of *n*).

In order to perform a download to a multivariable transmitter, you must make all of the slots INACTIVE that are associated with the multivariable transmitter. If you attempt to download before all of the multivariable transmitter's associated slots are INACTIVE, the following error message appears:

```
COMMAND ALLOWED ONLY IF ALL
PVS INACTIVE ON THIS MULTI-PV TRANSMITTER
```

Database discrepancies

Each slot indicates database discrepancies associated with that slot's parameters. To resolve smart transmitter database discrepancies, do the following:

Step	Action
1	Perform an upload.
2	If an upload does not resolve the discrepancies, perform a download.

Upgrading STI IOP to STIMV IOP

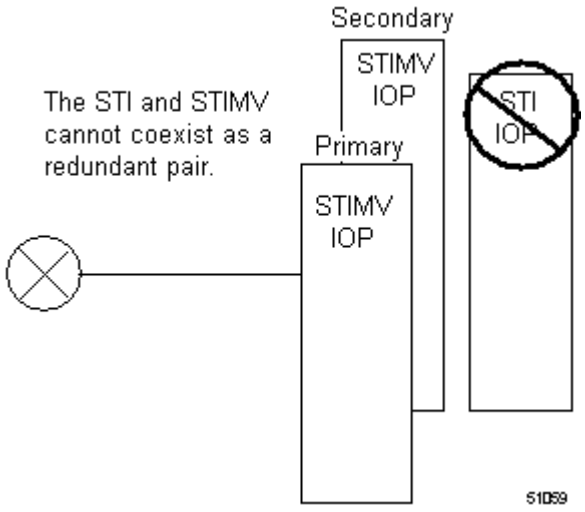
The STIMV IOP is a direct replacement for the STI IOP. Due to differences between the two types of IOP, certain precautions must be taken when upgrading the STI IOP to the STIMV IOP

"**Smart Transmitter Interface**" is the descriptor on the card edge of the **STI IOP**.

"**Smart MV Transmitter Interface**" is the descriptor on the card edge of the **STIMV IOP**.

CAUTION

In a redundant configuration where a STIMV IOP is used, both IOPs must be STIMV IOPs. A hot failover upgrade of a STI IOP to the new STIMV IOP is allowed; however, the upgrade procedure should be used to **replace both primary and secondary** STI IOPs with STIMV IOPs. Failover from a primary STIMV IOP to a secondary STI IOP is not permitted.



STIMV redundancy restriction

Synch error

The IOP box on the Status Display indicates an illegal redundant IOP configuration with a red `S` and red `BKP`. The `S` indicates a database synchronization error.

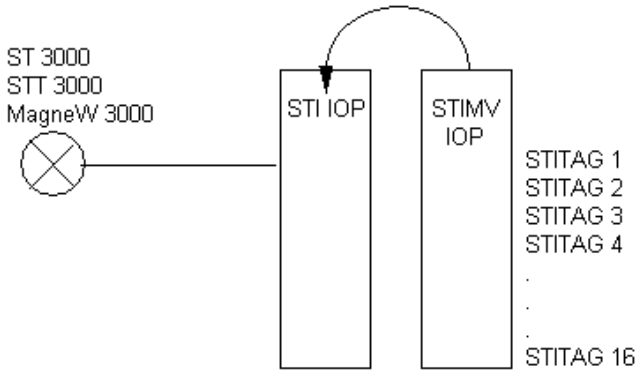
When a STIMV is the primary and an STI IOP is the secondary, the secondary database will not synch to the primary database, preventing redundant STI and STIMV operation. Replacing the secondary with a STIMV IOP clears the synch error.

ATTENTION

Before replacing an STI IOP with a STIMV IOP, ensure that all STITAG parameters for ST 3000, STT 3000, or MagneW 3000 transmitters associated with the IOP are unique. Performing this step prevents certain database discrepancies from occurring following the upgrade.

If you allow adjacent slots to have identical STITAG parameters, the STIMV IOP interprets the PVs as being from a single multivariable transmitter and a slot configuration overlap occurs.

A slot configuration overlap occurs because the PV data sent by the transmitters overlaps the database of what the IOP believes is a single transmitter database. To resolve slot configuration overlaps, the user should change the STITAG parameters to reflect the IDs of the transmitters connected to the IOP.



Check STITAG before upgrading STI with STIMV

Appendix B - Hierarchy building

This section includes information that you can reference for:

Topic	Link
Hierarchy building overview	Click here
Hierarchy tree views	Click here
CM/SCM containment	Click here
Parameter projection overiew	Click here
Using the Projected parameters tab	Click here
Connecting projected parameters	Click here
Using projected parameter for user templates	Click here
Block operations involving projected parameter	Click here
Reviewing Substitute name dialog box	Click here
Importing/exporting	Click here

Hierarchy building overview

Hierarchical building allows the creation of a Hierarchical View of a process from top down, or bottom up. The Hierarchy may reflect the Plant Data Model.

Containment View, in Control Builder's tree view, shows the containment relationship among all configured CM(s), SCM(s) as well as all basic blocks contained by the CM or SCM containers. The Assignment View will show the Assignment relationship among all blocks and basic blocks contained by it's container.

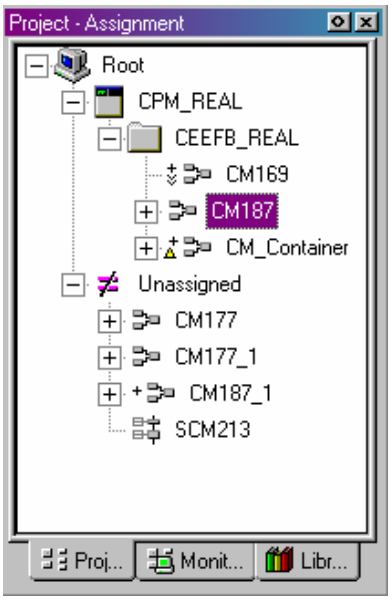
Parameter Projection allows the projection of parameters of an embedded block upward onto its container. When this container CM/SCM is embedded in another CM container, the projected parameters can then exposed as pins on the faceplate of this embedded block in the new container, and connections can be graphically made to these pins. Projected Parameters are defined as if they are the inherent parameters and will have all the attributes of the origin parameters, but will be referenced by their new names. The actual connections are made to the inherent origin parameters.

Hierarchy tree views

To show the views of both assignment and containment relationships among the configured blocks in the Experion system, two tree views are created: Assignment View and Containment View. To switch between these two views select Project/Monitoring tree and make the selection using a right mouse click of the View menu. The title bar of the tree view indicates the current view.

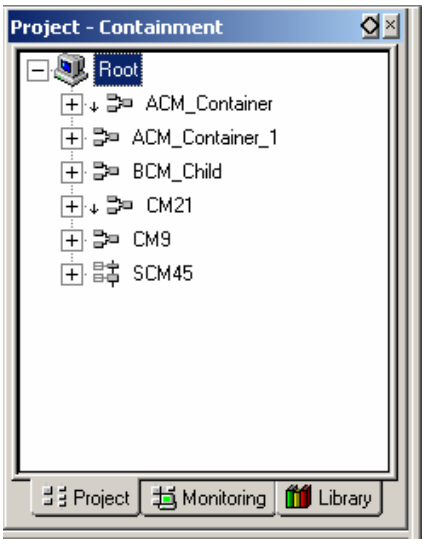
Assignment view

Assignment View shows the blocks involved in assignment as well as basic blocks contained by the containers. All unassigned blocks will be placed under Unassigned group rather than under the root. CM(s)/SCM(s) contained by a container CM will be shown under its assignment parent in this view rather than under its container CM. A "+" sign is used to indicate that the associated CM is a container.



Containment view

Containment view shows only the CM(s)/SCM(s) and contained blocks. The contained blocks can be CM(s), SCM(s), and basic blocks.



Because Containment Tree View doesn't show the assignment relationships among the configured blocks, some blocks (for instance, CEE, DEVICE, etc.) won't appear on the Containment Tree View. As a result, the following operations will not be available on Containment Tree View:

- Checkpoint
- Device Match/Unmatch
- Device Replacement

CM/SCM containment

This section includes information that you can reference for:

Topic	Link
Functional description	Click here
Containing CM/SCM	Click here
Uncontaining CM/SCM	Click here
Uncontain blocks with Projected Parameter(s)	Click here
Error scenarios for containment/uncontainment operations	Click here
Operations of container and contained CM(s)/SCM(s)	Click here
Load/load with contents	Click here
Performance restrictions	Click here

Functional description

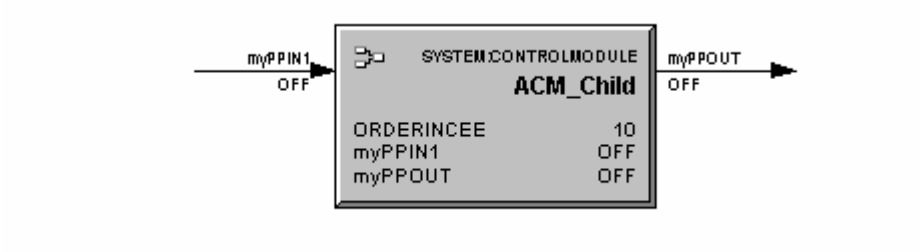
As a major part of the hierarchical building, CM/SCM containment allows the user to contain CM(s)/SCM(s) into another CM. The contained CM(s)/SCM(s) will remain as a tagged object, and as a result, the operation of containment does not affect the contained block's execution. In other words, the contained CM/SCM will still execute in its assigned CEE. Containment only encapsulates a potentially complex strategy and allows the user to define the process view based upon the actual data flow.

There are several ways to contain/uncontain CM(s)/SCM(s). However, neither operation will change the assignment parent of the intended CM(s)/SCM(s); the selected CM(s)/SCM(s) will remain unassigned if it is not assigned before the containment/uncontainment operation.

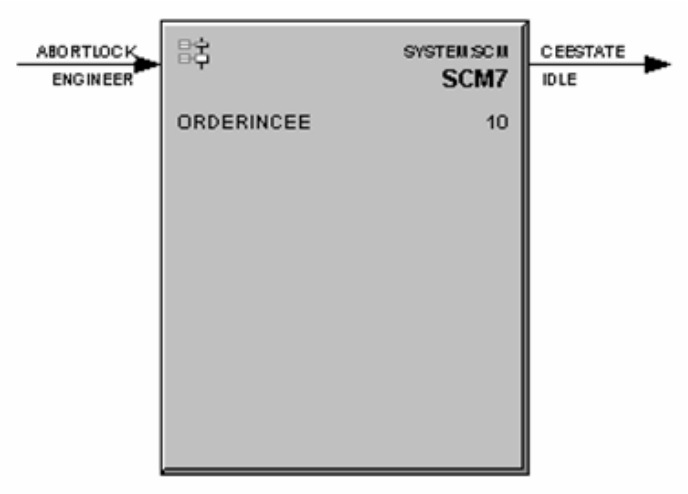
Containment operation alone won't cause delta flag for both the dropped block and the container CM. Only when the connections are made will the delta flag be set for the associated blocks.

Containing CM/SCM

In a CM container chart, a contained CM(s)/SCM(s) is indicated by a containment graphic on the left uppermost corner of the block's faceplate, as illustrated in the following figures.

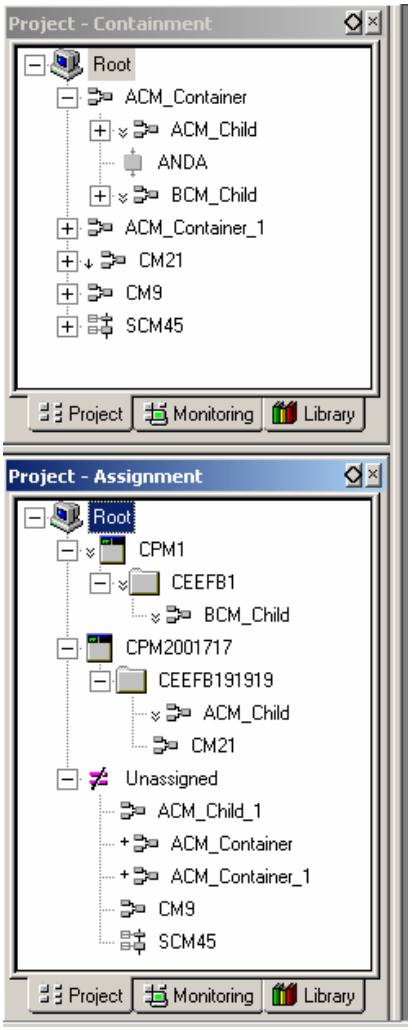


Block symbol of a contained CM



Block symbol of a contained SCM

The containment relationship does not require the contained blocks to be assigned to the same execution environment. In the figure below, the container CM, ACM_Container, itself is unassigned, but its containment children, ACM_Child and BCM_Child, are assigned to CPM2001717 and CPM1 respectively. In the assignment view the container CM, ACM_Container, is indicated by a "+" icon next to the name of the block on the tree.



Containment and assignment tree views

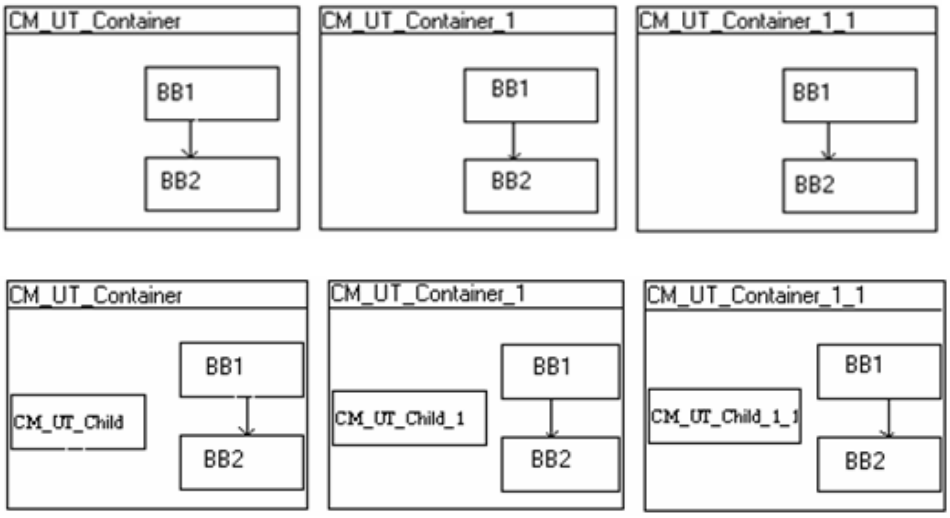
There are two possible ways to perform a containment operation, Drag and Drop and Module Containment Dialog.

After a user template is contained by a container user template, the contained user template will be specialized from the source user template. The propagation of this

contained user template will cause all derived container user templates and instances to have a derivation version of this contained CM.

For example, all CMs in the illustrations below are user templates. Before containment:

- CM_UT_Container_1 is derived from CM_UT_Container and
- CM_UT_Container_1_1 is derived from CM_UT_Container_1.



User template containment

After a user template UT_Child is contained into CM_UT_Container, CM_UT_Container has a specialized CM_UT_Child contained, all derivation children of CM_UT_Container, CM_UT_Container_1 and CM_UT_Container_1_1, are propagated with this new contained block.

An error will occur if an attempt is made to delete CM_UT_Child_1 or CM_UT_Child_1_1 because they all become parent defined and the user templates can't be deleted. The error will appear as "This operation is not valid at this level for a parent-defined element."

Drag and drop

In order to contain CM(s) or SCM(s) into another CM, drag the intended blocks from any project tree view and drop onto a CM in project Containment View or a project CM chart, even if the intended CMs/SCMs are already contained. CM blocks in the Assignment View cannot be the destination of a containment operation because the View only reflects the assignment configuration model of Experion system.

The containment parents cannot be changed for loaded contained CM(s)/SCM(s) unless these blocks are deleted from monitoring side first. The containment operation must be on the project tree.

The following table lists all possible drag and drop operations for containment. The Source column indicates the starting point of the dragged block, and the Destination column indicates the view or chart where the dragged block is dropped.

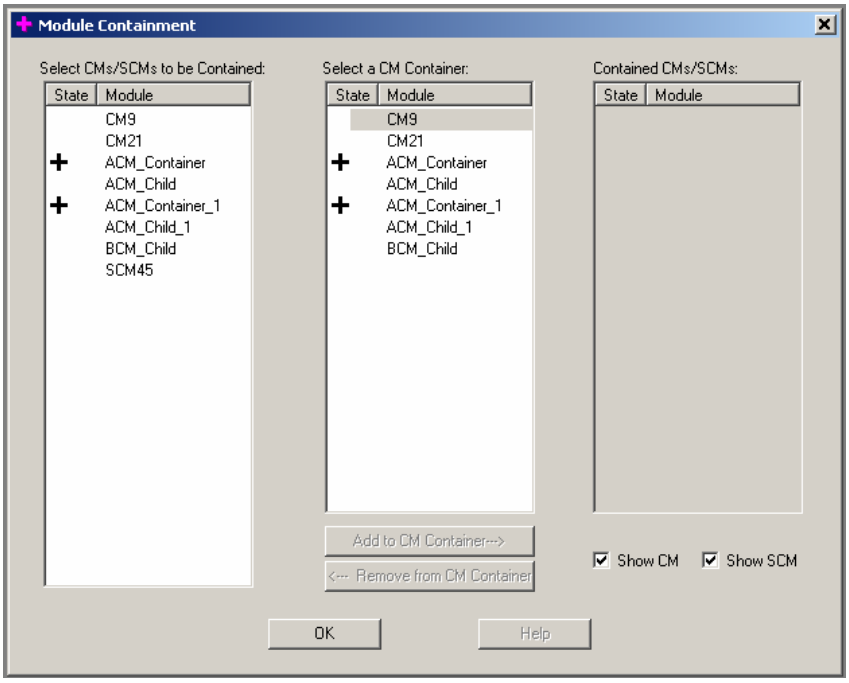
The dragged items can only be CM(s) and SCM(s) and the destination blocks are CM(s) only. Neither operation is supported in a Monitoring tree.

Table 2 Drag and drop operations for containment

Source	Destination
CM/SCM in Assignment View	CM in Containment view
CM/SCM in Assignment View	CM Chart
CM/SCM in Containment view	CM in Containment view
CM/SCM in Containment view	CM Chart
CM/SCM system profile or user template from the Library Tab	CM in Containment view
CM/SCM system profile or user template from the Library Tab	CM Chart
CM/SCM system profile or uncontained user template from the Library Tab	A CM User-Template in the Containment view, Library Tab
CM/SCM system profile or uncontained user template from the Library Tab	A CM user-template chart

Module Containment dialog

Module Containment Dialog may be used to contain multiple CMs/SCMs into another CM. A container will be indicated by a "+" sign.



Module Containment dialog

When the container CM in the middle section is selected, the right section of the dialog will display all contained CM(s)/SCM(s) by the selected container CM. Select Add to CM Container.

Uncontaining CM/SCM

Uncontaining CM/SCM undoes a containment operation. This operation applies to only control strategies.

There are three possible ways to perform an uncontainment operation, Drag and Drop, Module Containment Dialog and Uncontain menu option.

Drag and drop

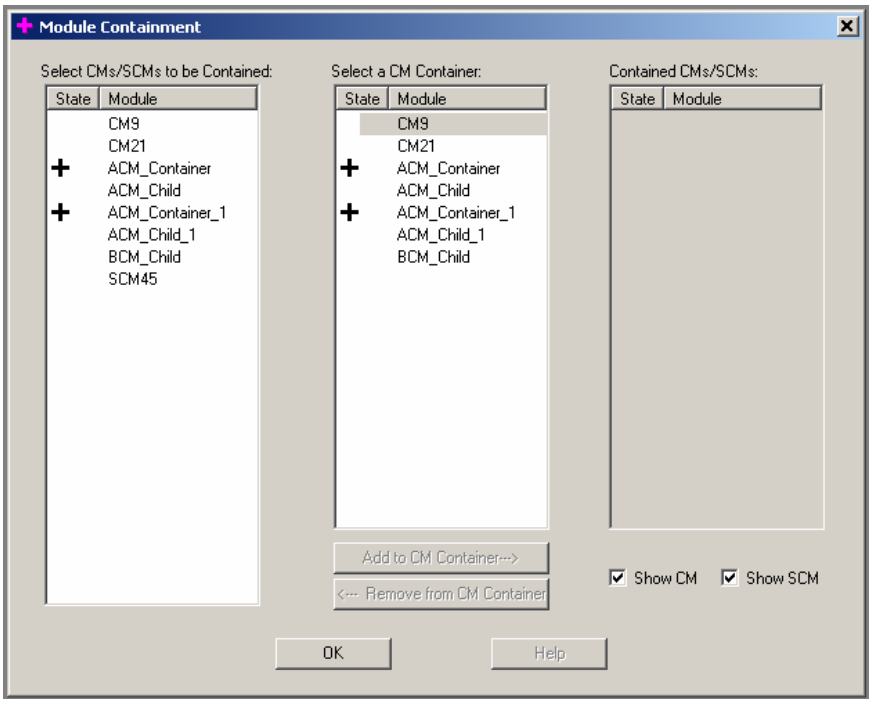
To uncontain using drag and drop, drag the CM/SCM from any project tree view or CM chart and drop into another CM in project Containment View. In this case the previously contained block will be uncontained from its previous containment parent and contained by a new container CM. The selected contained block may also be dropped onto the root of the Containment Tree View.

Drag and drop operations for uncontainment

Source	Destination
CM/SCM in Containment view	CM in Containment view
CM/SCM in Assignment View	CM in Containment view
CM Chart	CM in Containment view
CM/SCM in Containment view	CM Chart

Module Containment dialog

Module Containment Dialog also be used to uncontain multiple CM/SCMs into another CM. A container will be indicated by a "+" sign.



Module Containment dialog

When the container CM in the middle section is selected, the right section of the dialog will display all contained CM(s)/SCM(s) by the selected container CM. Select Remove from CM Container.

Uncontain menu option

To uncontain a CM/SCM within a CM chart, right click the item to be uncontained and select Uncontain. The selected CM/SCM may not be loaded. If the CM/SCM is loaded the menu item will not be available.

Uncontain blocks with Projected Parameter(s)

Two rules apply when uncontaining blocks with projected parameter and blocks with connections:

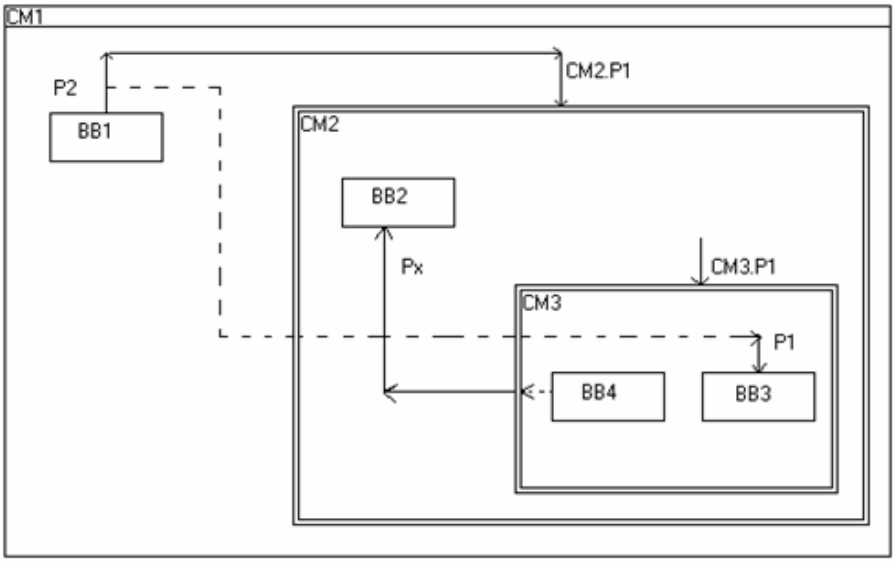
- Contained CM/SCM cannot be deleted if it is defined as an origin block;

- Uncontain will be prevented if the CM/SCM to be uncontained has graphical connections.

Rule examples

In this example, CM3 is contained by CM2 that is contained by CM1; P1 is a projected parameter that is defined as projected parameter on CM1, CM2 and CM3. The origin parameters for CM2.P1, CM3.P1 are defined as CM3.P1 and CM3.BB3.P1 respectively.

When making a connection between CM2.BB1.P1 and CM2.P1, an obvious graphical connection is created. But there is also a read only parameter connection created between CM1.BB1.P2 and CM3.BB3.P1 where CM3.BB3.P1 is the absolute origin for the Projected Parameter CM2.P1.



Uncontain Blocks with Projected Parameters

mmm) When uncontaining CM3, an error will be displayed,

"CM3 cannot be uncontained because CM3.BB3.P1 is defined as an origin parameter".

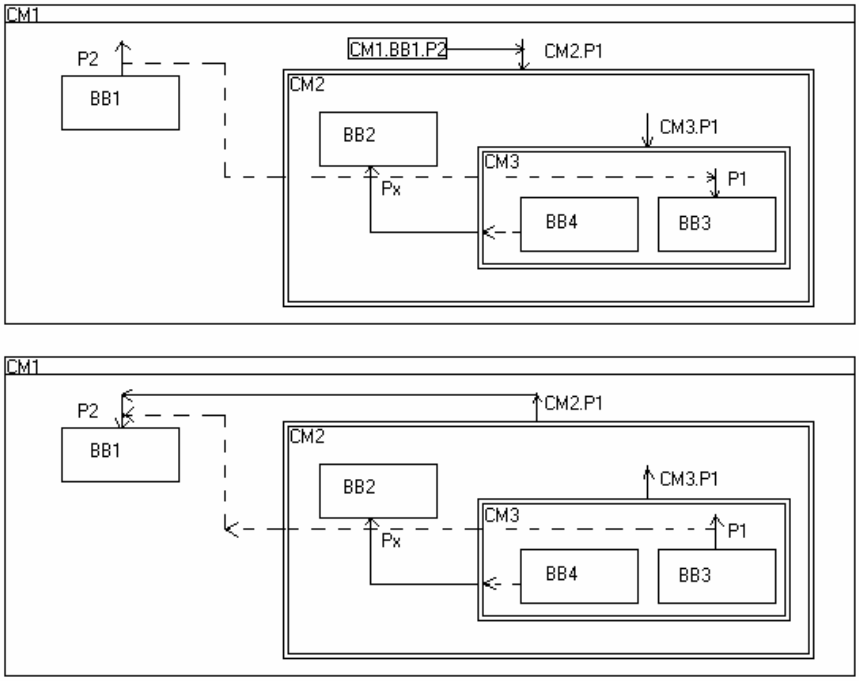
In order to uncontain CM3, the user needs to redefine the origin or empty the origin for CM2.P1 so that CM3 is not an origin block.

nnn) When uncontaining CM2, an error will be displayed,

"CM2 cannot be uncontained because there is a graphical connection to its projected parameter(s)".

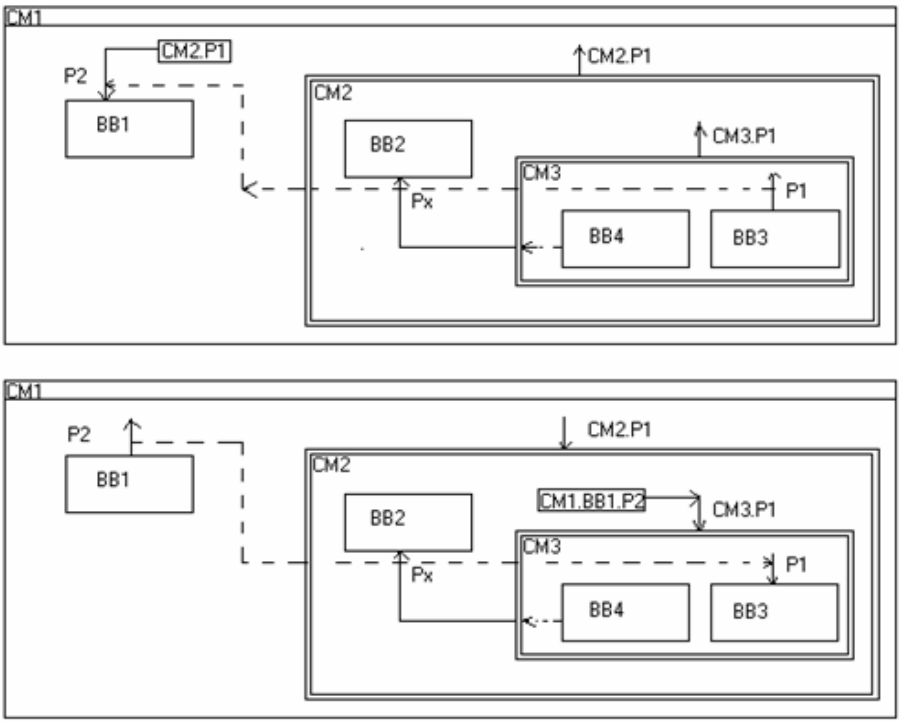
Deleting the connection between CM1.BB1.P2 and CM2.P1 first will allow the user to proceed with the uncontaining operation.

The same error will occur when uncontaining CM2 if there is a parameter connector on CM2.P1 whose source is CM1.BB1.P2, in which case CM2.P1 is a passive end of a graphical connection as indicated by the charts below.



Uncontain blocks with Projected Parameters

ooo) If CM2.P1 is used as a parameter connector to CM1.BB1.P2 however, as indicated by the chart below, because CM2.P1 is the passive end of a connection from a parameter connector, uncontaining CM2 will be allowed and the parameter connector will remain after the uncontaining CM2 operation.

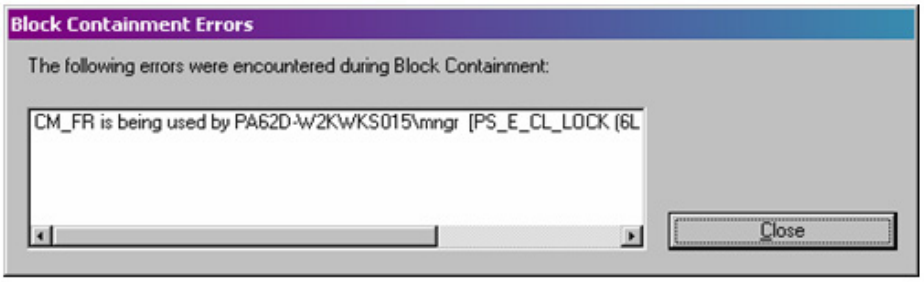


Uncontain Blocks with Projected Parameters

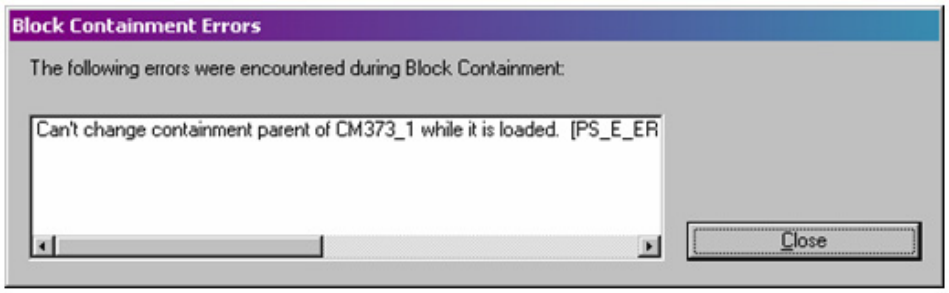
ppp) If CM1.BB1.P2 is used as a parameter connector to CM3.P1 as indicated by the chart above, uncontaining CM2 is allowed.

Error scenarios for containment/uncontainment operations

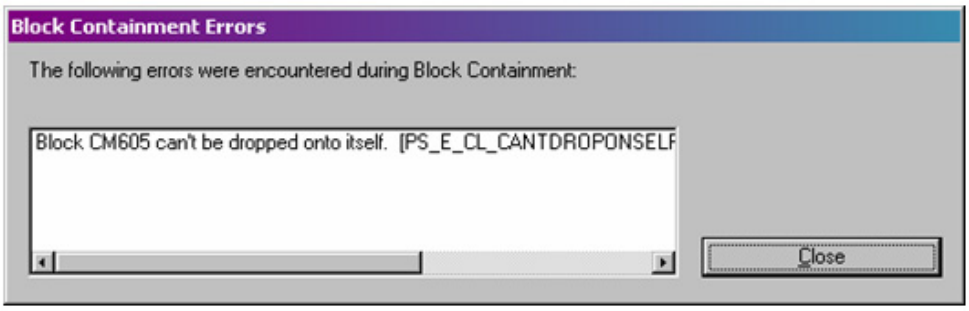
Regardless of the method used to contain or uncontain control modules, both the source and destination blocks will be locked by the containment operation. If the involved blocks are already locked by other operations, an error message will be displayed:



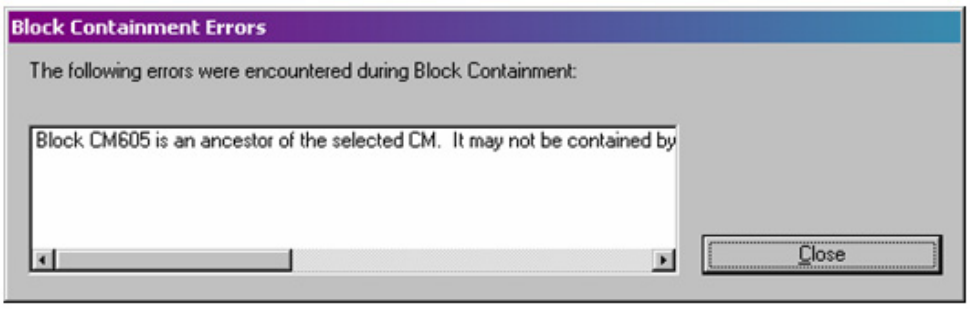
If the control module is loaded, its containment parent cannot be changed and an error will occur:



When using Module Containment Dialog, if the control module is selected to be both source and destination of the containment operation, the following error will occur: (this operation is prevented for drag and drop operation)



If a container CM selected is contained by one of its contained blocks, the following error will occur:



Operations of container and contained CM(s)/SCM(s)

Even after a CM(s)/SCM(s) is contained by a container CM, it remains a tagged object, with all the attributes of a tagged object. For instance, the contained CM(s)/SCM(s) can be loaded individually and has its own server point, its own execution state, etc. However, because it is contained by a container CM, some new operations are also supported.

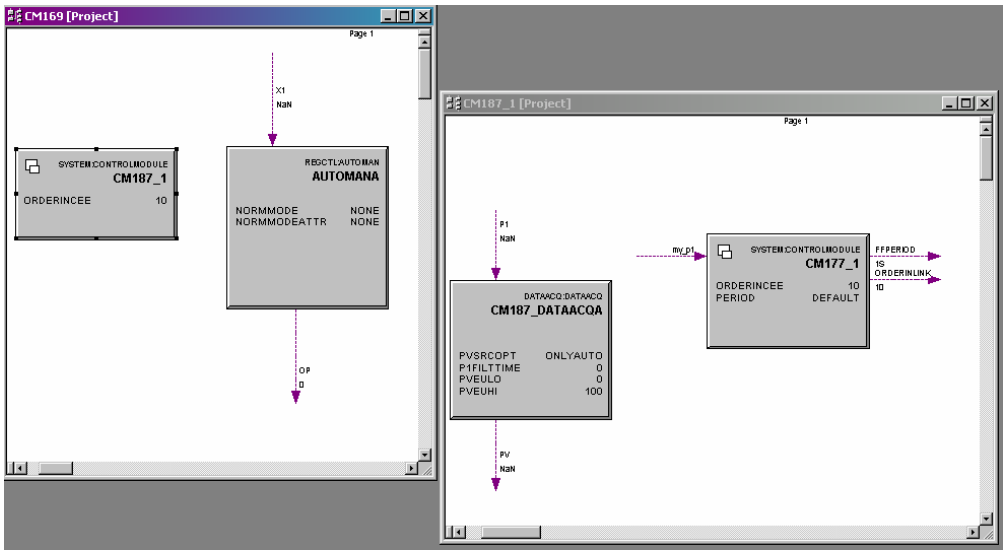
Editing a container or contained CM/SCM

The mechanism of editing a container CM or contained CM/SCM from tree view remains the same, assuming that they are not locked by other transactions.

Editing a contained CM/SCM from its container chart can be done in two ways:

Edit operations	Paths
Edit the form of a contained block	Double click on the contained block
	Right mouse click on the contained block and select Block Propertiesmenu item.
	Select the contained block and select Edit-> Block Properties
Edit the chart of a contained block	Right mouse click on the contained block and select Open Chart.

The assumption is that the contained block is not locked by another edit session. However, if the contained block is locked by another edit session, depending on the sequences of the operations involved, different error messages will be displayed to indicate the errors.



Contained CM/SCM edit example

Using the previous figure as an example, the following table lists the potential errors that may occur as a result of wrong sequence of operations and the ways to resolve the problems.

Editing errors

Scenarios	Operation	Error message	How to resolve
1	Step 1: Open the chart of container CM169; Step 2: Move or make connections to the contained block CM187_1 within CM169 chart; [At this time, CM187_1 is locked by CM169 transaction] Step 3: Double click on CM187_1 faceplate	CM187_1 is locked by CM169, please save CM169 and try again	Save CM169 chart and double click on CM187_1 [Save CM169 will unlock CM187_1 so that a new transaction can be started by CM187_1]
2	Step 1: Open the chart of container CM169; Step 2: Double click on CM187_1 to open its chart; [At this time, CM187_1 is locked by its own transaction] Step 3: Move, delete, uncontain or make connections to the contained block CM187_1 within CM169 chart	CM187_1 is locked by xxxx User, please resolve the problem and try again	Close the chart of contained block CM187_1

3	Step 1: Open the chart of container CM169; Step 2: Move or make connections to the contained block CM187_1 within CM169 chart; [At this time, CM187_1 is locked by CM169 transaction] Step 3: Double click on CM187_1 on the tree or right mouse click on CM187_1 on the tree and select Configure Module Configuration	CM187_1 is locked by CM169, please save CM169 and try again	Save the chart of container CM169 [Save CM169 will unlock CM187_1 so that a new transaction can be started by CM187_1]
4	Step 1: Open the chart of container CM169; Step 2: Double click on CM187_1 to open its chart; [At this time, CM187_1 is locked by its own transaction] Step 3: Double click on CM187_1 on the tree or right mouse click on CM187_1 on the tree and select Configure Module Configuration	CM187_1 is locked by xxxx User, please resolve the problem and try again	Save the chart of container CM169 [Save CM169 will unlock CM187_1 so that a new transaction can be started by CM187_1]

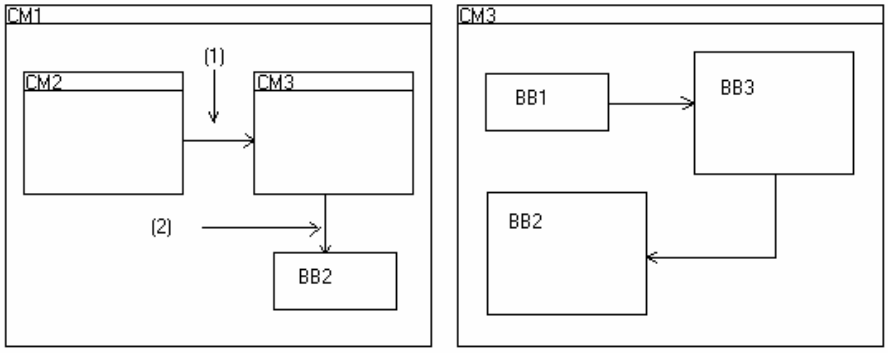
Normally, if a block is loaded and its configuration has been changed, a delta flag will be added next to the block's icon on the tree to indicate that there are changes. To resolve the differences, the user needs to reload the block from Project Tree. However, in Hierarchy Building, new rules are introduced to avoid unnecessary load if the changes are related to contained CM(s)/SCM(s),

- Changes made to a contained CM/SCM won't affect its container CM, therefore, no delta flag will be added for the container CM

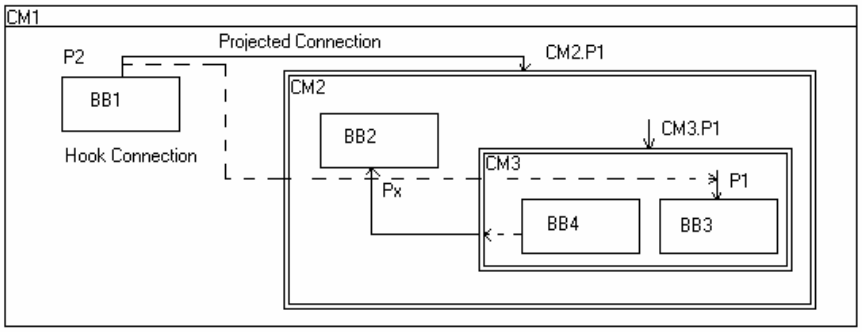
- Graphical changes, such as blocks being moved within the chart, won't cause any delta flag

Example 1: In this example, CMx indicates control module containers and BBx indicates basic blocks.

When making the connection (1), there will be delta flag for CM3 but no delta flag for CM1. When making connection (2), there will be delta flags for both CM1 and CM3 (this will happen if there is a back initialization connection). Any changes made within CM3 won't cause delta flag for CM1.

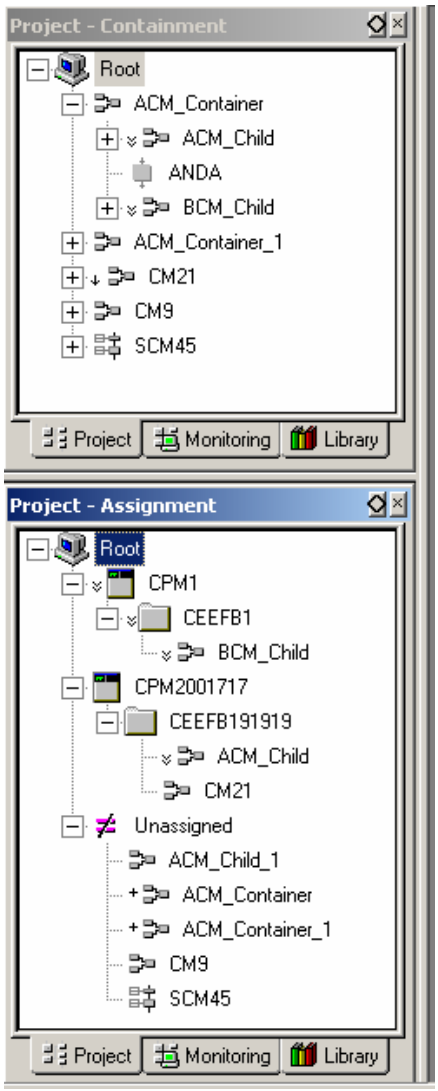


Example 2: In this example, CM3 is contained by CM2 that is contained by CM1; P1 is a projected parameter that is defined as projected parameter on CM1, CM2 and CM3. The origin parameters for CM2.P1, CM3.P1 are defined as CM3.P1 and CM3.BB3.P1 respectively. Making the connection between CM1.BB1.P2 and CM2.P1 will cause a delta flag for both CM1 and CM3, but not CM2 as a read only connection will be made between CM1.BB1.P2 and CM3.BB3.P1.



Assign/unassign a container or contained CM/SCM

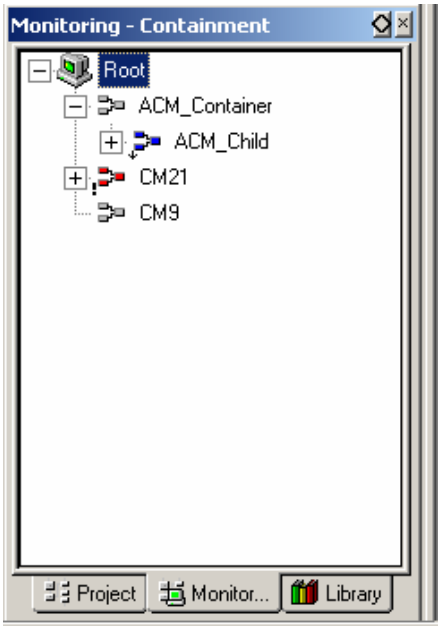
The containment operation does not have any impact on the assignment/unassignment of both the container CM or contained CM(s)/SCM(s). They can all be assigned to different execution environments.



Example for assign/unassign

In this example, ACM_Container contains ACM_Child and BCM_Child that are assigned to CPM2001717 and CPM1 respectively. CM_Container itself is unassigned. If CM_Container contains just ACM_Child, BCM_Child and no basic blocks, CM_Container does NOT have to be assigned in order to be loaded. In this case,

ACM_Child and BCM_Child will execute in their respective execution environments and CM_Container will not have an execution state, therefore, on the monitoring tree, the icon of CM_Container will be gray as showed below. However, CM_Container will become a server point and can be accessed in server.

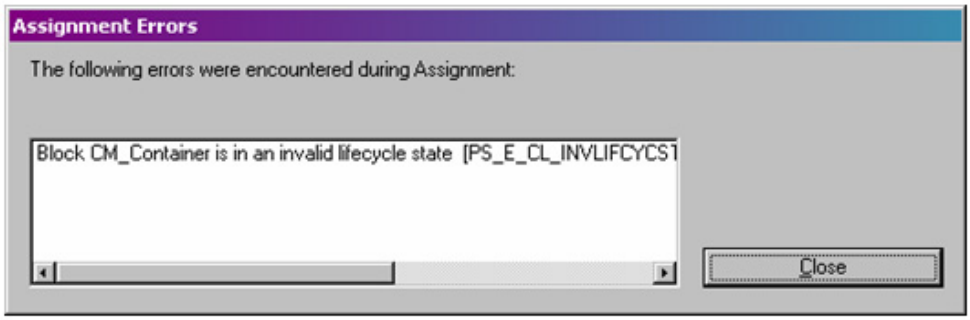


Example for auto load

Because CM_Container is unassigned and does not contain any basic blocks, it will be automatically loaded when either ACM_Child or BCM_Child is loaded. But loading CM_Container does NOT cause ACM_Child and BCM_Child to be loaded automatically. The user could also load CM_Container manually from Containment View. Again, because CM_Container is unassigned and has no execution state, the icon for CM_Container on the monitoring tree will be gray.

However, if CM_Container contains basic blocks, it has to be assigned to a control execution environment in order to be loaded.

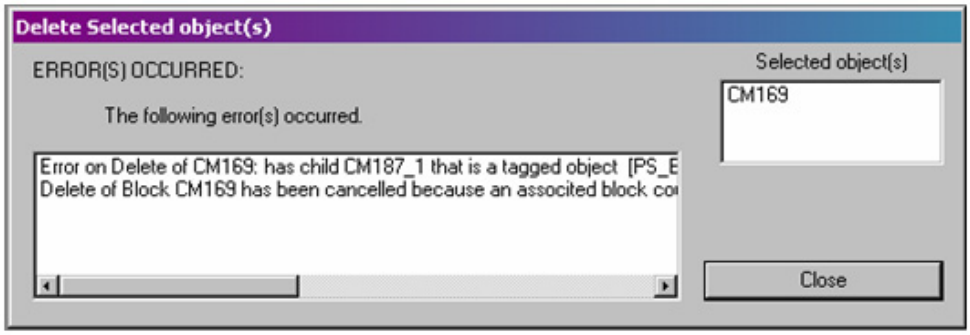
If the contained CM(s)/SCM(s) or container CM is already loaded, it cannot be assigned or unassigned, this is consistent with the existing restriction that a block cannot be reassigned if it is already loaded. The following error will occur to indicate the error,



An unassigned CM can be contained by another CM, and projected parameters can be configured on this unassigned CM. When contained by another CM, this unassigned CM behaves the same as any other contained CM/SCM, except that it doesn't have an execution environment.

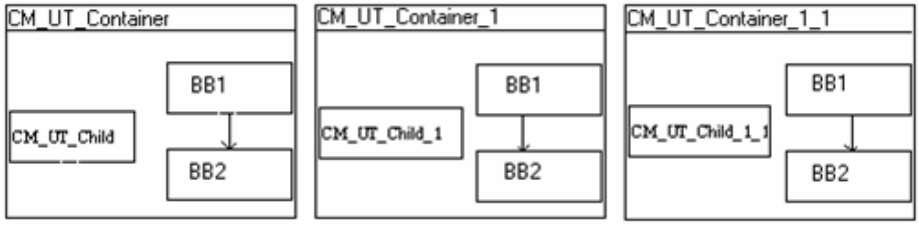
Delete a container or contained CM/SCM - deletion without Projected Parameter

The deletion of a container CM requires that all contained CM(s)/SCM(s) blocks be deleted first, this is consistent with the existing behavior where all tagged blocks must be deleted before an assignment parent can be deleted. An error message will be displayed if the deletion is attempted,



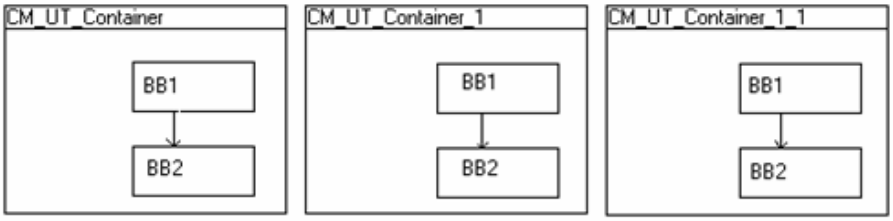
If a container CM block together with its contained CM(s)/SCM(s) blocks is selected to be deleted, the contained CM(s)/SCM(s) will be deleted programmatically before the container CM to comply with the rule specified above.

For user template, if the embedded user template has only embedded derivation children, the deletion of this embedded user template will delete all its embedded derivation children.



Example 1 for user template deletion

In this example, CM_UT_Container_1 is derived from CM_UT_Container and CM_UT_Container_1_1 from CM_UT_Container_1 and CM_UT_Child1, CM_UT_Child_1_1 and CM_UT_Child are template defining. All derivation children as well as instances of derivation children of CM_UT_Child are contained by other user templates. In this case, the deletion of CM_UT_Child will delete all its derivation children, CM_UT_Child_1 and CM_UT_Child_1_1.



Example for 2 user template deletion

If the embedded user template to be deleted has non-embedded derivation children, or non-embedded instances (in other words, derivation children or instances are not contained by other user templates or control modules) the deletion will be prevented. In the above example, if CM_UT_Child has a non-embedded derivation child CM_UT_NONEMBED_Child, deletion of CM_UT_Child will cause an error,

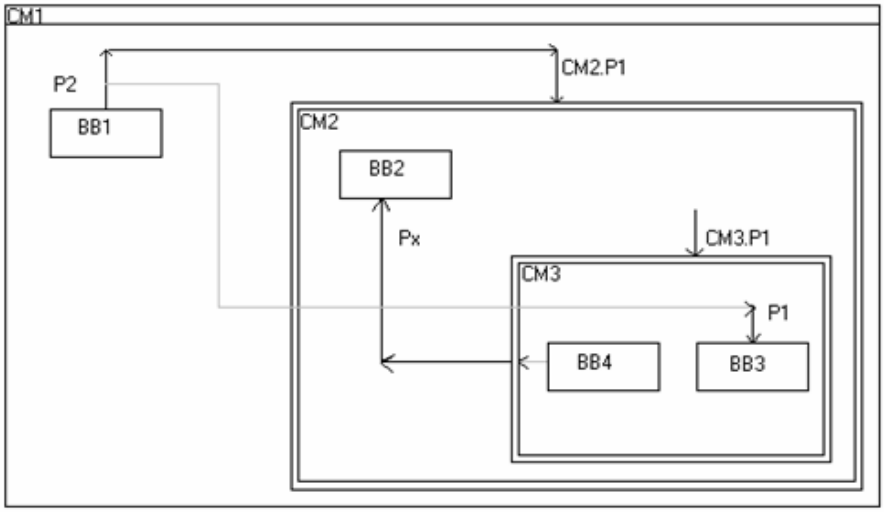
"The selected block CM_UT_Child has non-embedded children block CM_UT_NONEMBED_Child, please resolve the problem and try again."

CM_UT_ NONEMBED_Child must be deleted before CM_UT_Child can be deleted. And the deletion of this user template will delete all its derivation children.

Delete a container or contained CM/SCM - deletion with Projected Parameter

In this example, CM3 is contained by CM2 that is contained by CM1. P1 is a projected parameter that is defined as projected parameter on CM1, CM2 and CM3. The origin parameters for CM2.P1, CM3.P1 are defined as CM3.P1 and CM3.BB3.P1 respectively.

When making the connection between CM1.BB1.P2 and CM2.P1, besides the graphical connection, there is also a read only connection made between CM1.BB1.P2 and CM3.BB3.P1 because the CM3.P1 as the origin of CM2.P1, itself is a projected parameter.



Example for block deletion involving Projected Parameter

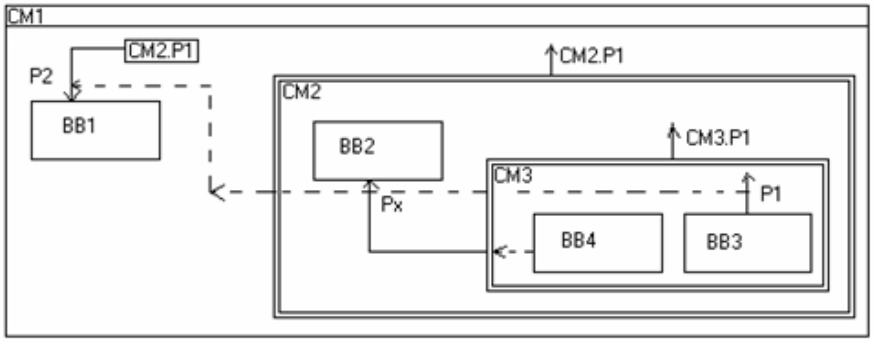
3. When deleting CM3, an error will be displayed,

   "CM3 cannot be deleted because CM3.BB3.P1 is defined as an origin parameter".

   In order to delete CM3, the user needs to redefine or empty the origin of CM2.P1 so that CM3 is not a origin block. If both CM2 and CM3 are selected for deletion, the operation will be allowed because the parent is in the scope of the operation.

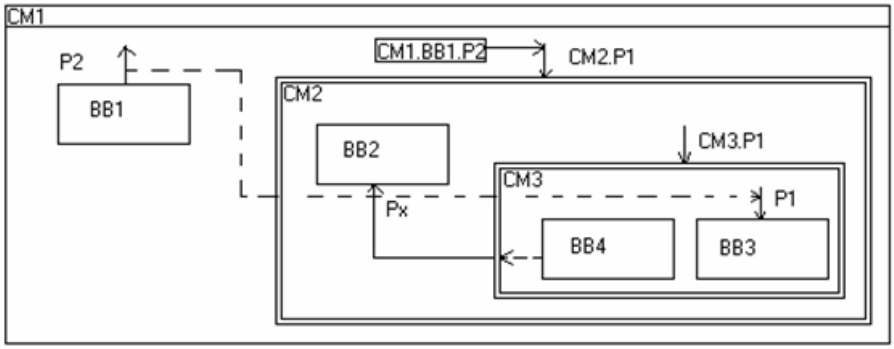
qqq) When deleting CM2, the graphical connection between CM1.BB1.P2 and CM2.P1 as well as the read only connection between CM1.BB1.P2 and CM3.BB3.P1 will be deleted.

rrr) If CM2.P1 is configured as parameter connector to CM1.BB1.P2, deleting CM2 will cause the connection to CM1.BB1.P2 dangling. "???????" will be shown in the parameter connector box.



Example for block deletion involving Projected Parameter

sss) If CM1.BB1.P1 is configured as parameter connector to CM2.P1 as indicated by the chart below, deleting CM2 will cause the parameter connector to CM2.P1, CM1.BB1.P1 deleted.



Example for block deletion involving projected parameter

Delete a container or contained CM/SCM - deletion of the loaded container or contained CM/SCM from the loaded side

If a container CM is deleted from monitoring, all monitoring instances of the contained CM/SCM's will be placed under the root. This will break up parameter projection chains under projected connections but the corresponding read-only connections will stay. If a container CM is deleted from monitoring, all dependent basic blocks will be deleted as well. If the container CM is reloaded, the projection chain will be re-established and the contained CM/SCM will be underneath the container CM.



TIP

CMs can be deleted individually from the loaded side regardless of whether they are contained or container. However, container CMs cannot be deleted from the projected side unless all contained CMs/SCMs are deleted first.

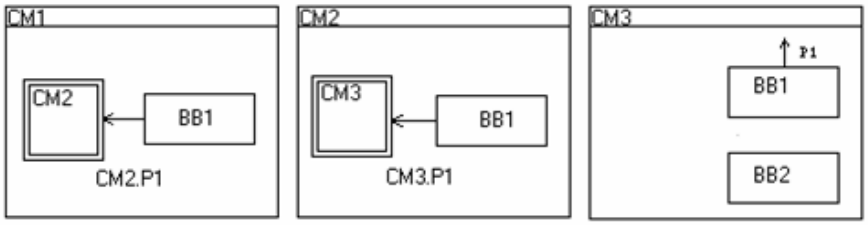
Copy a container or contained CM/SCM

When a container CM is copied, all contained CM(s)/SCM(s) are copied in addition to the dependent blocks. Unique names will be assigned for the contained CM(s)/SCM(s), all named connections and the origins of projected parameter will be resolved.

Projected parameters will be maintained on copy, for instance, if CM1 has a projected parameter with origin CM1.PIDA.PV, after copy, the new CM1_1 will have a projected parameter named CM1_1.PIDA.PV.

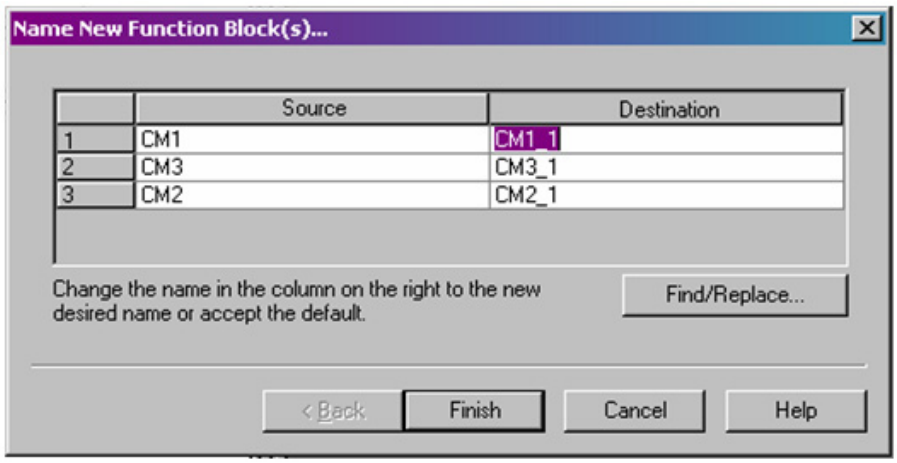
The existing mechanism for resolving Substitute Name and Substitute Connections will be invoked if they are involved in the copy operation. If a projected CM/SCM or container has projected parameter defined on it and is copied, the projected parameter definition and any projected connections are copied to the new block or container.

In this example, CM3 is contained by CM2 that is contained by CM1. P1 is a projected parameter that is defined as projected parameter on CM1, CM2 and CM3. The origin parameters for CM2.P1, CM3.P1 are defined as CM3.P1 and CM3.BB3.P1 respectively.

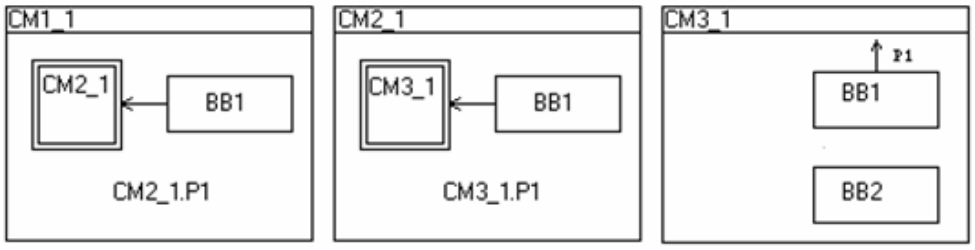


Example 1 for block copy

When copying CM1, the following dialog will be displayed so blocks involved in the operation can be renamed.



If no changes are made, the new blocks as well as their projected parameter will look like the following after the copy operation,

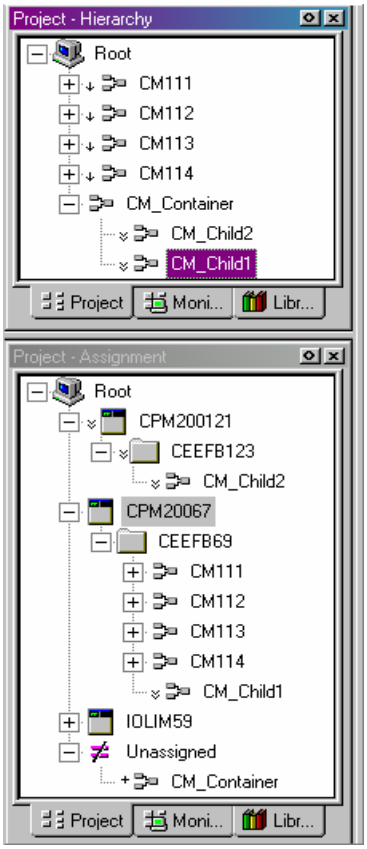


Example 2 for block copy

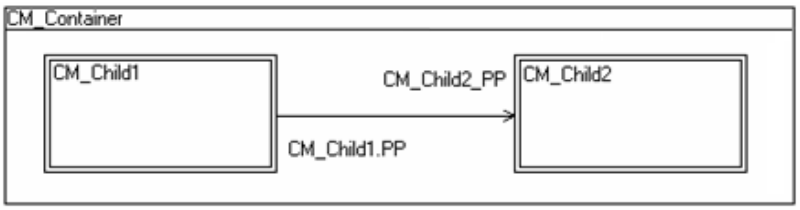
Load/load with contents

Even if a CM/SCM is contained by a container CM, the contained CM/SCM remains to be a tagged object and can be loaded without its container CM being loaded first. The contained CMs/SCMs can in fact be executed in different execution environments.

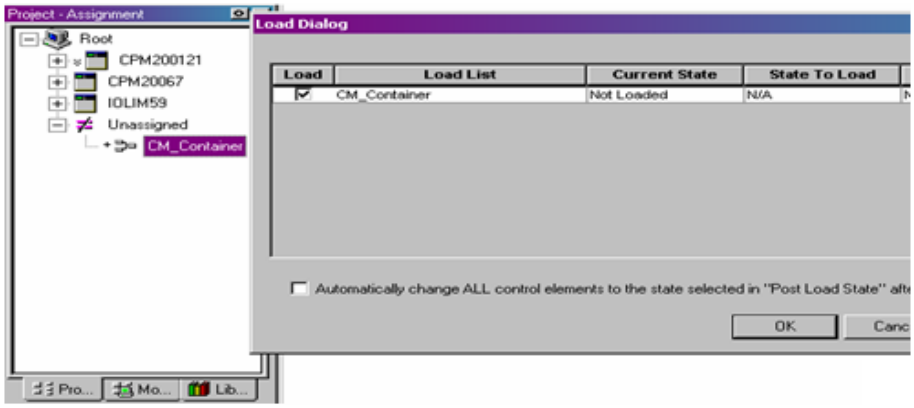
The user is able to load an unassigned container CM if it doesn't contain any basic blocks and its contained blocks are properly assigned. This unassigned container CM will have its own server point, therefore can be accessed in Server. Once loaded, the unassigned CM will appear on the Monitor Tab of the Containment view. The icon is gray because the unassigned CM does not have an execution state. Because this CM is not assigned and there isn't an Unassigned category for Monitoring Tab, this CM will not be found in Assignment Tree View.



In the example, the unassigned CM_Container contains just two control module blocks CM_Child1 and CM_Child2.

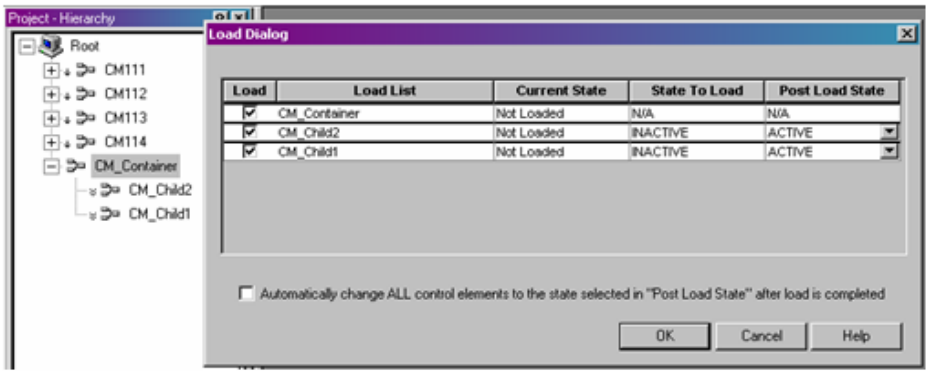


If a container CM_Container is selected for Load with Contents, depending on which tree view the load is originated, two different sets of contents will be selected for loading. If Load with Contents for the container starts from Assignment Tree View, the load list contains only the selected container CM_Container. In the example, the results of Load and Load with Contents of CM_Container will be identical if the operation starts from the Assignment Tree View as shown in the figure below.



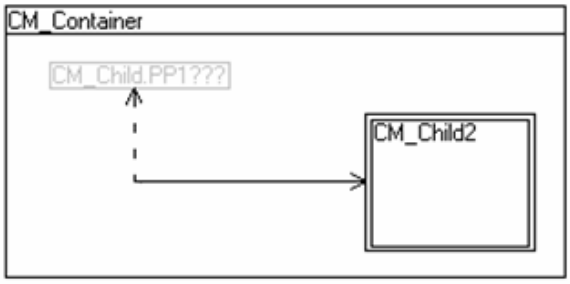
Load Dialog from assignment tree view

However, if Load with Contents is originated from Containment Tree View, both the container CM and all contained CM(s)/SCM(s) are selected for loading.



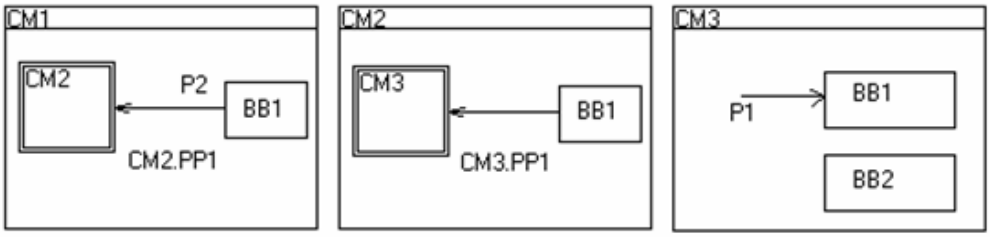
Load dialog from containment tree view

As stated earlier, if an assigned CM is to be loaded and it's container CM is unassigned (and has no basic block) and has not been loaded, the container CM is automatically loaded. In this example, CM_Child2 is selected to load and CM_Container will be loaded automatically. After the load, the CM_Container chart on the monitoring side will look like this:



Because CM_Child1 is not loaded, there is a read-only parameter connector indicating the other end of the connection.

Another example involves partially loading the hierarchies. In this example, CM3 is contained by CM2 that is contained by CM1. PP1 is a projected parameter that is defined as projected parameter on CM1 and CM2; P1 is a projected parameter defined on CM3. The origin parameters for CM2.PP1, CM3.PP1 are defined as CM3.PP1 and CM3.BB1.P1 respectively.



If only CM1 and CM3 are loaded, the real connection between CM1.BB1.P2 to CM3.BB1.P1 will be loaded to the controller. However, since CM2 is not loaded, on the loaded side, the CM1 chart will not have CM2 contained. A warning will be displayed to indicate that the containment parent for CM3, that is CM2 is not in the load list.

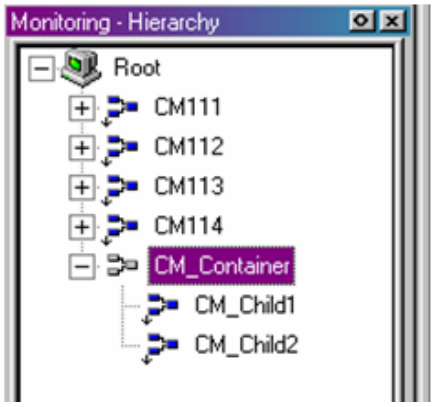
If the user loads only CM1 and CM2, because the origin block CM3.BB1 is not loaded, the "real" connection between CM1.BB1.P2 to CM3.BB1.P1 will not be loaded to the controller.

After the CM is loaded, the configured projected parameters on this CM can be accessed in Server just as any other inherent parameters. These projected parameters can also be used in server history configuration and sever scriptors.

Activate/inactivate

The contained CM(s)/SCM(s) can be activated/ inactivated as independent blocks and the activation/ inactivation of the container CM has no impact on the activation/inactivation of the contained CM(s)/SCM(s). Likewise, containment does not affect the alarms generated by each CM/SCM regardless whether the selected tagged block is a container CM or container CM/SCM.

However, the container CM can be selected and "Activate this Item and its Contents" or "Inactivate this Item and its Contents" option from the Containment Tree View can be used to activate/inactivate both container CM and contained CM(s)/SCM(s) at the same time. However, if the container CM_Container were unassigned, as illustrated below, the container CM_Container does not have an execution state, therefore, only Activate/Inactivate Selected Item(s)' Contents menu option is exposed. Selecting the menu item will cause CM_Child1 and CM_Child2 to be activated or inactivated (if they are active).



Upload

The containment of CM(s)/SCM(s) has no impact on the upload of both the container CM or contained CM(s)/SCM(s) if they are all assigned. If the container CM is not assigned, Upload operation will be disabled. Upload with Contents only apply to CMs with all contained CMs assigned.

Update to project/update with contents to project

As independent blocks, the container CM as well as contained CM(s)/SCM(s) can be selected individually for Update to Project operation. If the container CM is selected in Containment Tree View of the monitoring tab, Update with Contents (to Project) allows the user to do the update for both container and contained blocks at the same time. In the example below, Update to Project with Contents for CM1 will cause the contents for CM1, CM2 and CM3 updated to project. Update to Project for CM1 will only update BB1 contained by CM1 and its connections.

Once the contained CM/SCM is loaded, its containment parent cannot be changed. Therefore, the hierarchy cannot be changed once the contained CM/SCM is loaded. However, the basic blocks can be deleted from the control strategy without any restriction. In the example, if BB2 is deleted from CM3 on the project side and an Update with Contents for CM1 or Update to Project for CM3 is performed, a BB2 will be created and contained into CM3.

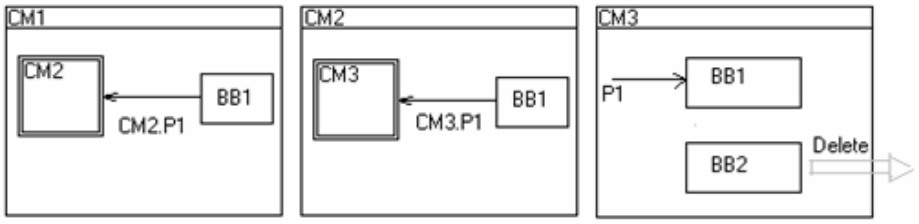
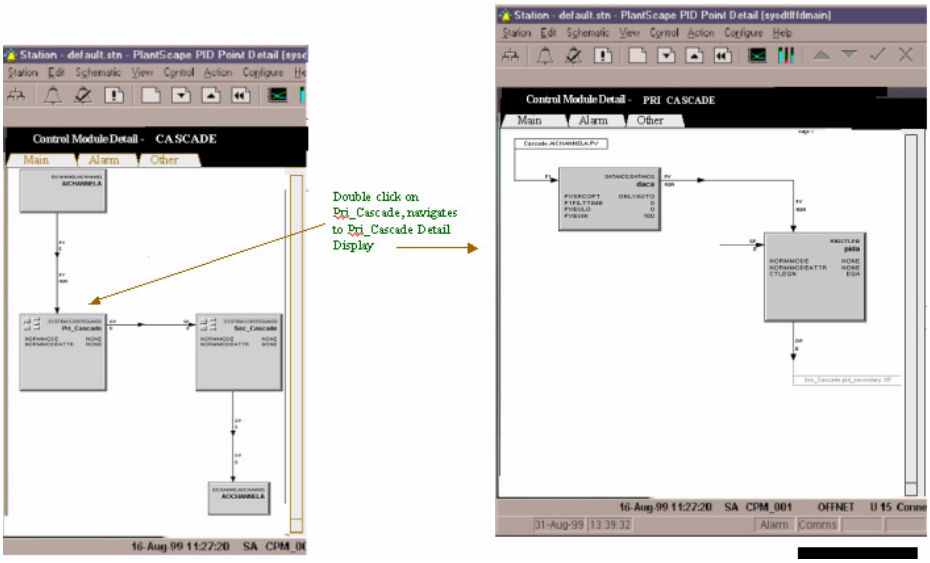


Chart visualization

Since a CM chart can be embedded in a Station Display using Chart Visualization, and multiple charts shouldn't be opened at the same time in Station Display, the purpose of right-clicking "Navigation to Source" is to navigate to the detailed display of the selected CM/SCM as if the name of the selected CM/SCM was typed and the F12 function key selected. Double-clicking the contained CM/SCM invokes the block's form.



Performance restrictions

The number of levels for CM/SCM containment is limited to 5.

Parameter projection overview

This section includes information that you can reference for:

Topic	Link
Naming and renaming	Click here
Original parameter overview	Click here
Empty original parameter	Click here
Assigning/unassigning/reassigning the original paramter	Click here
Validating the original parameter	Click here
Changing the value	Click here

Naming and renaming

The Parameter Projection function provides the ability to create a parameter on a block and assign the new parameter to another parameter in the system. The new user-defined parameter is called a ‘projected parameter’ because it projects the value and attribute from another parameter (called its ‘origin parameter’) onto itself.

The projected parameter will look and act just like an inherent parameter of the block on which it was defined. For instance, the projected parameter can be nominated as a Pin or block faceplate parameter just like an inherent block parameter can. It can also be accessed like any inherent parameter of the CM/SCM. But it will reflect the value and attributes of its assigned origin parameter. When a connection is made to the projected parameter, the connection, which is loaded to the controller, is really a connection made to the absolute origin parameter of the projected parameter.

Parameter projection also allows the projection of a parameter over multiple levels of containment, but this requires a projection of this parameter on each containment level. In other words, the projection of a parameter cannot skip the projection levels. The resulting projection chain will project the absolute origin over one or more intermediate levels of containment up to an upper level projected CM/SCM. Although the origin blocks can only be those blocks contained by the CM on which the projected parameter is defined, a projection chain can be modified when the origin parameter of any projected parameter that participates in the projection chain is deleted or redefined,

in which case all upwards connections associated with the modified origin will be deleted or remade.

Definition of Projected Parameter and Assignment of Origin Parameter apply to the Project side only. No configuration changes to the projected parameter can be made on the Monitoring side.

Common

The name of a projected parameter may be modified at any time during configuration and should follow the same naming conventions and rules for its uniqueness just as any parameter. It may be the same name as the origin parameter or some other user-meaningful name.

Common naming conventions for projected parameter

Projected Parameter	Description
Max string size	The string size of a projected parameter name must not exceed the maximum string size for alias and projected parameter names currently implemented as 255 char.
Name parts	A projected parameter name may be compound from up to three parts.
Name part separator	Parts must be separated by a dot.
Valid character	Each part must start with a alphabetic char and may be followed by an alphabetic or numeric character.
Valid examples	MyProjParam MyP123 P1.P2.P3 MyPP.Aaa.bbbbb
Invalid examples	‘123‘ - Name must start with an alphabetic character ‘MyPP1.P2.P3.P4’ - To many parts ‘MyPP1.123.P3’ - Second part doesn’t start with an alphabetic character ‘MyPP1.123-P3’ - Invalid character in second part

The table below describes the validation performed to confirm that the name of a projected parameter does not conflict with any other name in the system.

Projected parameter name validation

Projected Parameter name	Full name (unique identifier)	Validation
Part1	Tag.Part1	Check against other parameter: 'Part1' must not conflict with an inherent or another projected parameter of the CM/SCM. Check against basic blocks: Note: There is no reason to permit that Part1 is also used as a basic block name contained by the CM/SCM.
Part1.Part2	Tag.Part1.Part2	Check against other Parameter: 'Part1.Part2' must not conflict with an inherent or another projected parameter of the CM/SCM. Check against basic blocks: If a basic block named 'Part1' exists inside the CM/SCM then 'Part2' must not conflict with an inherent parameter name of this basic block.
Part1.Part2.Part2	Tag.Part1.Part2.Part3	Check against other Parameter: 'Part1.Part2.Part3' must not conflict with an inherent or another projected parameter of the CM/SCM. Check against basic blocks: If a basic block named 'Part1' exists inside the CM/SCM then 'Part2.Part3' must not conflict with an inherent parameter name of this basic block.

Validation

The following items in the system must be validated to avoid name confliction:

- Parameter name must not conflict with the name of an inherent parameter of the projected CM/SCM (as well as Instances and derivation children in case of user template).
- Parameter name must not conflict with an existing projected parameter of the projected CM/SCM (as well as Instances and derivation children in case of user template).
- Parameter name must not conflict with the name of a basic block contained by the projected CM/SCM (as well as Instances and derivation children in case of user template).
- Parameter name has to be checked as is currently done for aliases in terms of string size and use of valid characters.

Origin parameter overview

Once an origin is assigned, it defines the behavior of the projected parameter. The projected parameter may be used as if it is the absolute origin in terms of rules for connections and pin/faceplate exposure. Since the origin of a projected parameter can be a projected parameter itself, an origin can be projected multiple levels. Such capability will facilitate top-down design of control strategies.

Restrictions: Substitute parameter and alias parameters of SCM cannot be defined as the origin parameter.

Empty origin parameter

Empty origin is indicated by a symbol appended to the name of the projected parameter, allowing the configuration of a projected parameter without assigning an absolute origin. The external interface to a block can be defined and connected to its peers before anything about its internal implementation is defined.

In the case of user templates, because origin is not defined, it is not template defining. As a result, the user could define different origin to the same projected parameter for derivation children and instances. However, if the origin is defined as an existing parameter, for user template, the origin will be template defining and the origin will be propagated to all derivation children and instances.

In summary, empty origin will:

- Allow an projected parameter to be template defined but the corresponding origin parameter is not;
- Make it easy to fix broken projection chains, i.e. when projected CM/SCMs, which participate in a projection chain, are missing during import, the chain can be fixed by unassigning the missing origin parameter rather than failing the import operation;
- Provide top-down design capability.

Assigning / unassigning / reassigning the origin parameter

The origin parameter may be assigned, unassigned or reassigned at any time during configuration. The origin parameter can be redefined by clearing the existing origin.

Assignment, unassignment, or reassignment will affect the following:

Affected item	Results
Corresponding projected parameter	• Delta flag will be set for the projected CM/SCM. • The corresponding faceplate parameter value will be updated. • The corresponding block pin value will be updated.
Projected connections (graphical connections as well as parameter connectors)	• Delta flag will be set on the block that owns the projected connection.
Read-only connections	• Delta flag will be set on the block that owns the read-only connection.
Each projected parameter upwards the projection chain	• Delta flag will be set projected CM/SCM that owns the projected parameter. • The corresponding faceplate parameter value will be updated. • The corresponding block pin value will be updated.

Origin parameter assignment:

If a projected parameter with an empty origin is involved in connections, the assignment of a valid origin parameter will create read-only connections between the absolute origin parameter of the projected parameter and the other end of the connections.

Assignment of an origin parameter whose absolute origin is not defined will be allowed without any restrictions.

Origin parameter unassignment:

The unassignment of an origin parameter with a valid absolute origin will delete read-only connections formed between the absolute origin parameter and the other end of the projected connection. Performing unassignment needs validation to be sure that all connections can be deleted. If the validation fails the unassignment operation may not be permitted.

Origin parameter reassignment:

Origin parameter reassignment consists of two steps: clear the current origin parameter and define a new origin parameter. If the assignment validation or the unassignment validation fails, the reassignment operation will fail and the original origin parameter will remain.

Validating the origin parameter

Validation rules for the unassignment operation

Origin to be modified	Unassignment validation
Empty origin (origin not defined)	No validation needed (Rationale: the absolute origin is already/still unassigned)
Projected parameter of a contained block whose immediate origin is a projected parameter with absolute origin undefined.	
Absolute origin parameter from a dependent block.	For the corresponding projected parameter and all projected

Origin to be modified	Unassignment validation
Projected parameter from a contained block whose immediate origin is a projected parameter with absolute origin undefined.	parameter upwards the projection chain: The owning block can be locked successfully. For all existing projected connections, the read only connections on the absolute origin parameter can be deleted successfully.

Validation rules for the assignment operation

New origin	Assignment validation
Empty origin (origin not defined).	No validation needed (Because: the absolute origin is not specified)
Projected parameter from a contained block whose immediate origin is a projected parameter with **absolute origin undefined**.	
Absolute origin parameter from a dependent block.	For the corresponding projected parameter and all projected parameter upwards the projection chain: The owning block can be locked successfully. Exposed faceplate pins must match with symbol attributes of the absolute origin. For all existing projected connections, a read-only connection can be formed successfully on the absolute origin parameter.
Projected parameter from a contained block whose immediate origin is a projected parameter with absolute **origin undefined**.	

Assumes current origin is defined

An assignment operation will not necessarily define an absolute origin parameter, i.e.: when the new origin is an unassigned projected parameter.

Changing the value

The value of a projected parameter is the projected value of its absolute origin. When the the value of a projected parameter is changed on either project side or monitoring side, the value of the absolute origin parameter will be changed. If the validation for this operation fails, then the value change will not be permitted.

If the absolute origin for a projected parameter has not been defined, then the value will not be shown and cannot be changed.

Using the Projected Parameters tab

This section includes information that you can reference for:

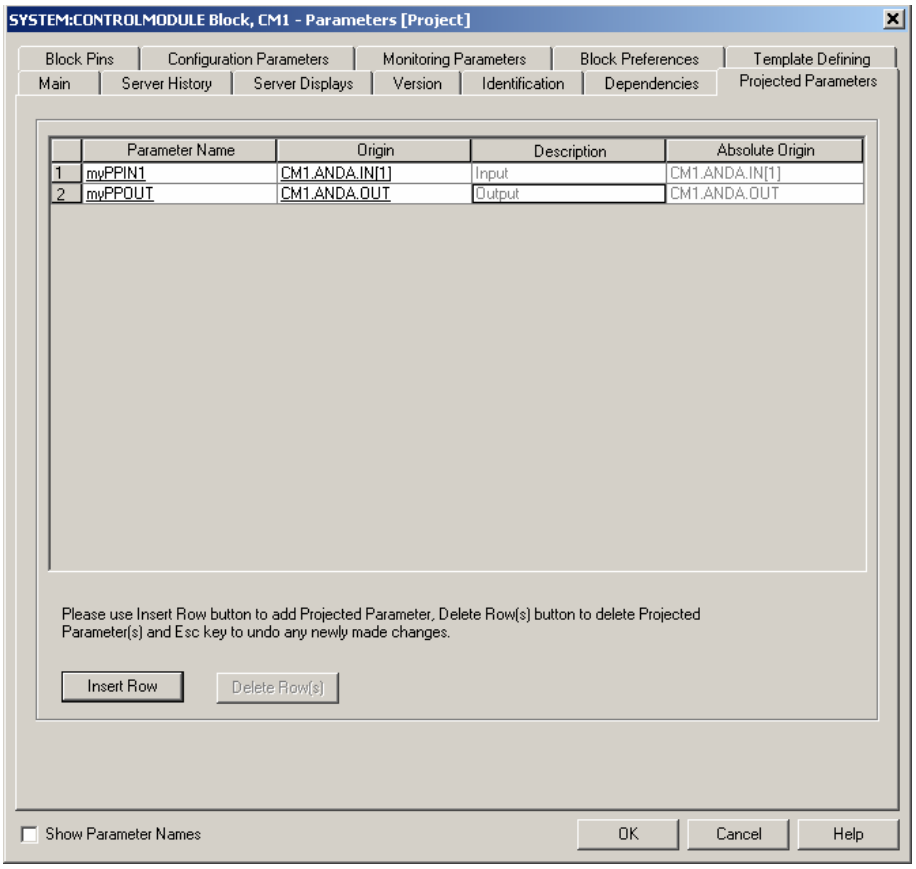
Topic	Link
Configuration form overview	Click here
Symbol attributes	Click here
Adding a Projected Parameter	Click here
Deleting a Projected Parameter	Click here
Printing Projected Parameters	Click here

Configuration form overview

The configuration form for projected parameter can be accessed by a tab in the Configure Module Parameter dialog box of the projected CM/SCM. The form displays a list of all defined projected parameters for the projected CM/SCM and the following information about each parameter:

- Name of the projected parameter
- Origin parameter
- Parameter description

The configuration form provides the ability to add, modify, and delete a projected parameter to the projected CM/SCM. All fields are read-only on the monitoring side.



Configuration form for Projected Parameters

Grid columns and cell validation

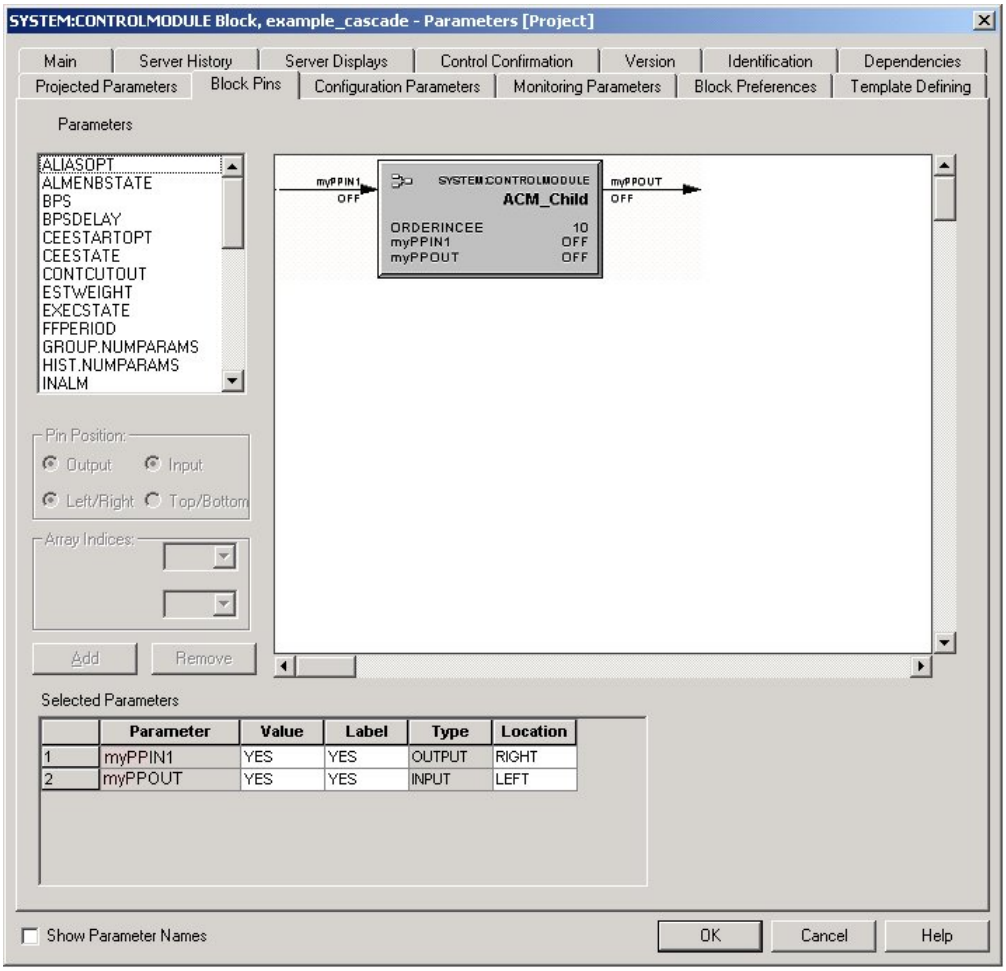
Column	Behavior	Cell validation
Parameter Name	The name field of the projected parameter is an **edit box** and shows the name of the projected parameter defined by the user. The ESCAPE key can be used at any time during the edit session to restore the entry for the current field.	The entry must follow the naming rules for projected parameter definition. If the validation fails, an error message will appear.
Origin	The origin parameter field will be an edit **box** with **point picker** access. An origin parameter name can be entered or the point picker invoked to select a parameter from a list of all system parameters. For multi-level of projection, the Origin shows the immediate origin of the projected parameter. The ESCAPE key can be used at any time during the edit session to restore the entry for the current field.	The entry must follow the validation rules for origin parameter. If the validation fails, an error message will appear. Upon successful validation, concerned connections will be reconnected and the parameter description field will be updated to show the description of the just assigned origin. If at this point the projected parameter name field is empty then a default parameter name will be assigned.
Description	The parameter description field is be a **static field** that is set and refreshed when the corresponding origin is shown or changed.	No validation required.

Column	Behavior	Cell validation
Absolute Origin	The Absolute Origin field is read-only. For one-level of projection, the Absolute Origin is the same as the immediate origin. For multi-level projection, Absolute Origin is the starting point of the parameter projection chain.	No validation required.

Symbol attributes

A Projected parameter is exposed as a pin on the faceplate of the projected CM/SCM so it can be graphically connected to another parameter. A projected parameter can also be exposed on the block's faceplate as a configuration or monitoring parameter. Once a projected parameter has been defined, it appears in the symbol attributes pages for selection as a faceplate pin or parameter where it is listed in the choice list along with the block's standard parameters.

In the following screen capture of the Block Pins Configuration Tab, the parameter myPPIN1 and myPPOUT are projected parameters defined on ACM_Child. These parameters have been exposed as pins on the ACM_CHILD block. The Configuration and Monitoring Parameters Tabs also have this projected parameter as a selection parameter in the left hand list box as a faceplate parameter.



Projected Parameters as block symbol attributes

The value of a projected parameter may change when:

- The value of its absolute origin is changed.
- The absolute origin has been changed, i.e. an origin at an intermediate level is reassigned.

If the value of a projected parameter is being changed, the system must verify:

- An absolute origin is defined.
- The absolute origin block can be locked successfully when the value change form opens.
- The value of the absolute origin parameter can be changed successfully.

Adding a Projected Parameter

To add a new projected parameter, select the Insert button. Edit the fields to define the name and origin of the newly defined projected parameter.

If the origin is defined, but the name field is left empty, a unique name for the projected parameter will be automatically generated.

Validation will take place when the cursor is moved to another control. If the validation fails, an error message will be displayed.

Deleting a Projected Parameter

Use the Delete key to delete:

- The projected parameter, that currently has the focus on the grid control
- Multiple projected parameter, if multiple rows have been selected in the grid control

Validation Rules for Deletion:

- The projected parameter must not be used as an origin for another projected parameter.
- The projected parameter must not be exposed as a pin on the faceplate.
- The projected parameter must not be exposed as a faceplate parameter.

If the validation fails, deleting the projected parameter is not permitted.

Printing Projected Parameters

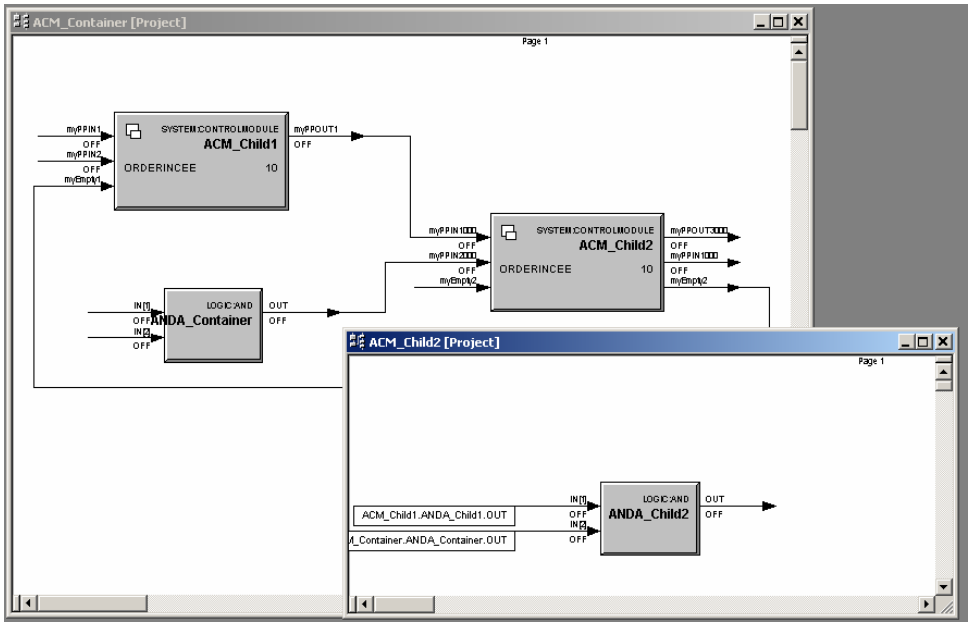
Projected Parameter appear in the printing reports for the projected CM/SCM in the parameters section as regular parameters, except it is indicated that they are actually projected parameters and the reports will display their corresponding origin parameter assignment, if any.

Alias Configuration	
ALIASOPT	OFF
Projected Parameters	
myPPIN1	ACM_Child.ANDA.IN[1]
myPPOUT	ACM_Child.ANDA.OUT
Connections	
-- No Connections --	

Connecting Projected Parameter

Projected parameters may be used to expose a pin on the faceplate of the projected CM/SCM that may be connected to another inherent parameter or projected parameter either graphically or via a parameter connector or expressions in calculation, step and transition blocks.

A connection that has a projected parameter at one or both ends always appears on the chart (or form) as it is created, with the projected parameter name. In all cases, a read-only parameter connector is displayed at the origin parameter to note that a connection has been formed to the origin parameter, as illustrated in the following figure. If the projected parameter is the passive end of a connection, a cross reference is displayed if the option is enabled. If a connection is made or deleted on a projected parameter, the delta flag of the blocks with the owning end of the absolute origin connection is set to indicate that an additional load operation is necessary to reflect the new connection to the controller. Projected parameter may participate in a connection as any parameter can.



Projected connection examples

Read-only connections

If a connection is made to a projected parameter with a defined absolute origin, an additional read-only connection is created between the absolute origins. Read-only connections can't be created between projected parameters or between projected parameter and inherent parameter.

The read-only connection can be loaded to the controller, but projected connections cannot. Making and deleting a read-only connection sets the Delta Flag of the block that owns the absolute origin connection, indicatiing that an additional load operation is necessarily if this block is in another CM/SCM.

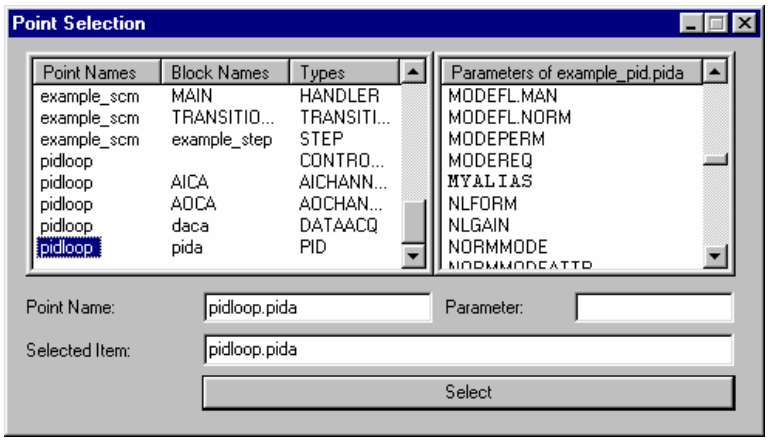
Validation

Connection	Validation
Connections with Projected Parameter when Absolute origins are not defined	If a projected connection is to be made and at least one end of the connection has no absolute origin defined, the connection will be allowed with the following validation: If the projected parameter has been exposed as a pin, a minimum validation will be performed, that is, an input pin can't be connected to an input pin and an output pin can't be connected to an output pin, regardless of what the origin parameter might be. Other connection rules are applied when the absolute origin is defined.
Connections with Projected Parameters, when Absolute origin is defined	If a connection is made to a projected parameter and the absolute origin of the parameter on both ends are defined, then the system must verify: A read only connection can be made successfully between the absolute origins using existing rules. If the validation fails, then the connection may not be permitted.

Point Picker

The Point Picker displays projected parameters in the parameter list along with the block's standard parameters.

In the following screen capture of the Point Picker tool, the parameter MYALIAS on the right is a projected parameter defined on pidloop.pida.



Projected Parameters in Point Picker

The Point Picker shows only the parameters of the contained blocks that include both contained basic blocks and contained CMs/SCMs.

Using projected parameter for user templates

This section includes information that you can reference for:

Topic	Link
Propagation	Click here
Scenarios and examples	Click here

Propagation

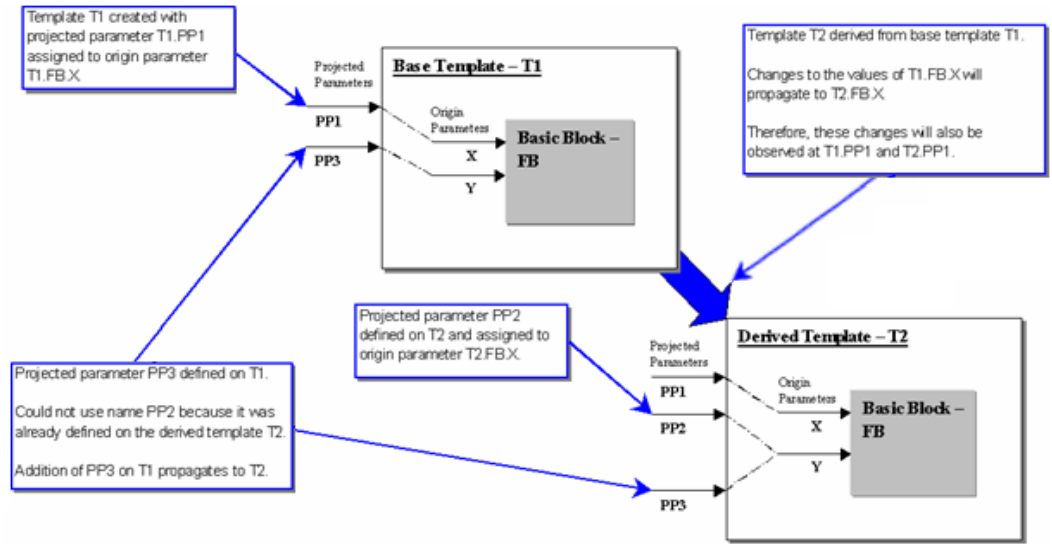
Projected parameter definitions are template defining and propagated from parent templates to derived templates and instances. Origin assignment for a projected parameter is template defining if origin is defined, and changes to the origin assignment in the template propagate to its derived templates and instances. Because symbol attributes propagate and are template-defining, the pin and faceplate projected parameter assignments automatically propagate to derived templates and instances.

Scenarios and examples

The following diagram is an example of projected parameters on templates. Template T1 is created and a projected parameter, PP1, is defined on the template. Then template T2 is derived from template T1, and it inherits projected parameter PP1 definition and its origin assignment to FB.X.

Next a projected parameter named PP2 is defined on the derived template T2 and assigned to FB.Y on T2. If another projected parameter is defined on T1, it cannot be named PP2 because projected parameter definitions are always template defining and will propagate to derived templates and instances.

Because of this conflict, the new projected parameter is named PP3 and assigned to origin parameter FB.Y. The origin parameter assignment propagates to the derived template T2. Before allowing the assignment, the system ensures that there is not a conflict on derived templates and instances. In this case, FB.Y is already assigned on the derived template T2, but it is not connected so there are no rules preventing the multiple assignment.



Projected Parameters in Templates

Block operations involving Projected Parameter

This section includes information that you can reference for:

Topic	Link
Loading block with projected connections	Click here
Update to Project	Click here

Loading block with projected connections

Loading of tagged block is allowed even if the origin blocks are not loaded. However, an error will occur if the to-be-loaded CM has projected parameters with origin undefined, and the load will fail.

Update to Project

The containment relationship can't be changed once the hierarchy is loaded, so Update to Project operation does not involve updates for containment relationship.

If a projected CM/SCM is updated to project, all its projected parameters are checked to ensure that the projection chain downwards to the origin is valid, which includes the validation of both origin parameters and projected connections.

If a projected connection is updated during the update operation, the corresponding read only connection is updated as well on the project side.

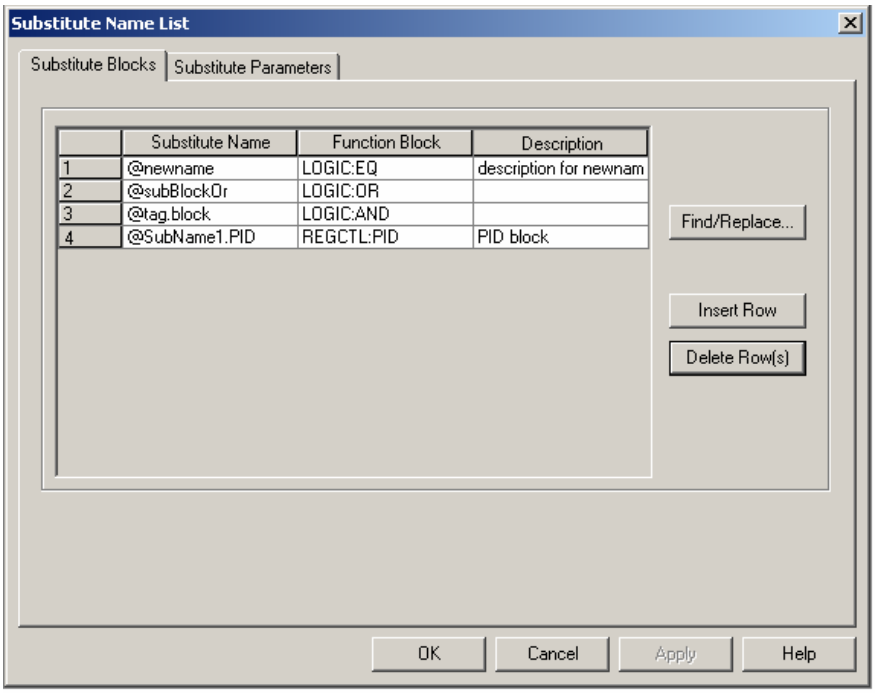
Read only connections are not updated to project directly. Only projected connections will be updated.

If Update to Project operation fails for any reason, the projected parameter will remain, but the origins of the projected parameter will be emptied.

Reviewing Substitute Name dialog box

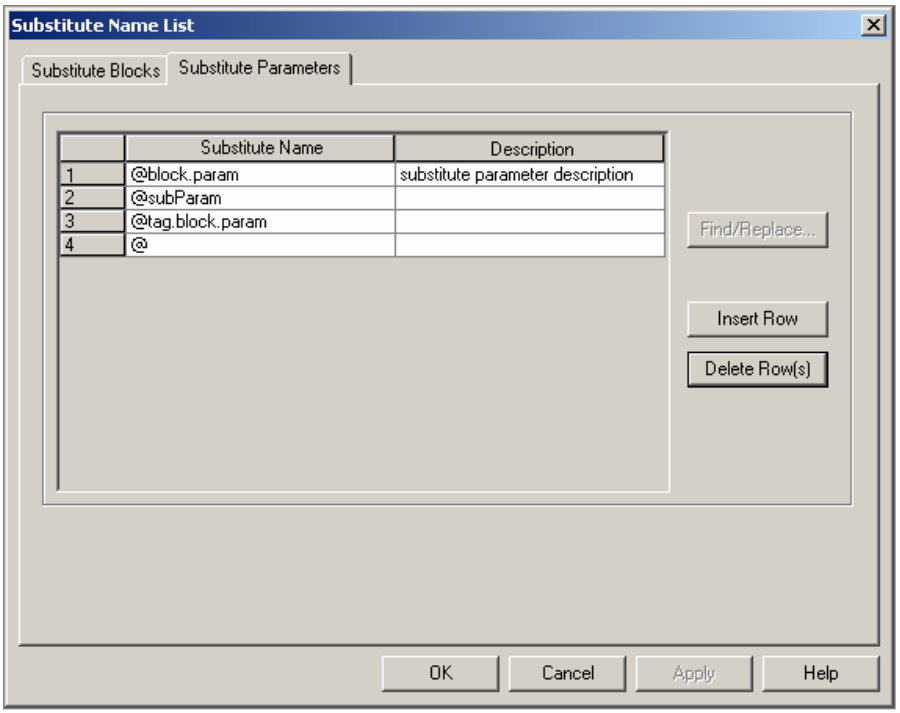
The Substitute Name dialog box is divided into two tabs with Substitute Blocks as the default tab.

The Substitute Blocks tab will contain a grid to assign substitute names for blocks.



Substitute blocks tab

The Substitute Parameter tab will have a grid to support the substitute names for parameters (substitute parameters). Substitute parameter names provide a placeholder for a parameter value; tag.parameter or tag.block.parameter, where a substitute block name provides a placeholder for a tag or tag.block.



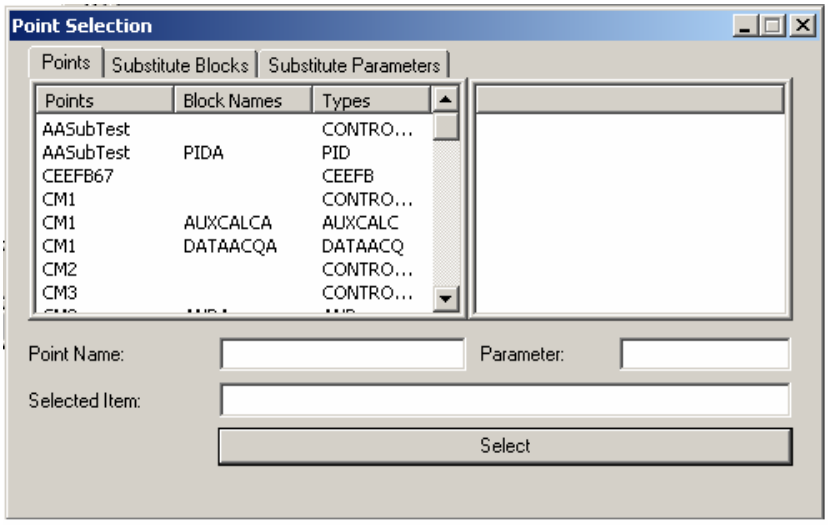
Substitute Parameters tab

Using the Point Picker

The Point Picker Dialog consists of three tabs:

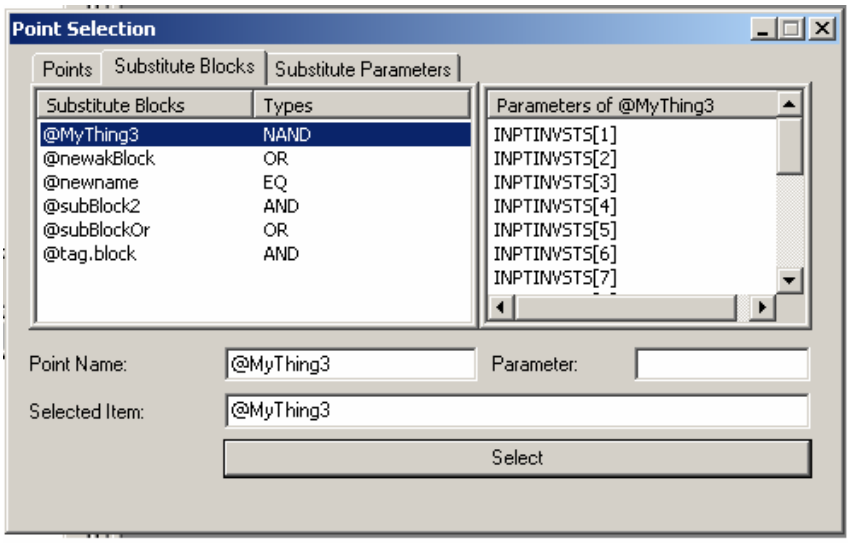
- Points (Standard Point Selections)
- Substitute Blocks
- Substitute Parameters

In the Points tab shown below, AASubTest is the point selected and its associated parameter choices.



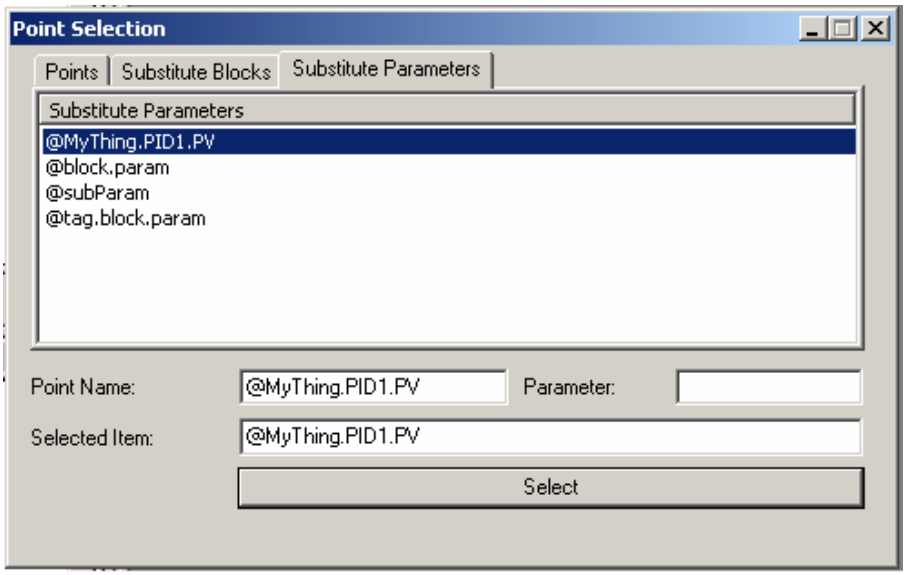
Points tab

The Substitute Blocks tab shows the substitute block names and their type. @MyThing3 is the substitute block selected and the associated parameter list for type PID.



Substitute Blocks tab

In the Substitute Parameters tab shown below, @MyThing.PID1.PV is a substitute parameter.



Substitute Parameters tab

Resolve substitute connections

The Resolve Substitute Connections dialog box supports substitute blocks and parameters. Substitute parameters must be assigned an origin before connections/names can be resolved. The Function Block column will read N/A for substitute parameters.

Importing/exporting

This section includes information that you can reference for:

Topic	Link
Containment	Click here
Projected Parameter	Click here
Strategy Import/Export dialog box	Click here

Containment

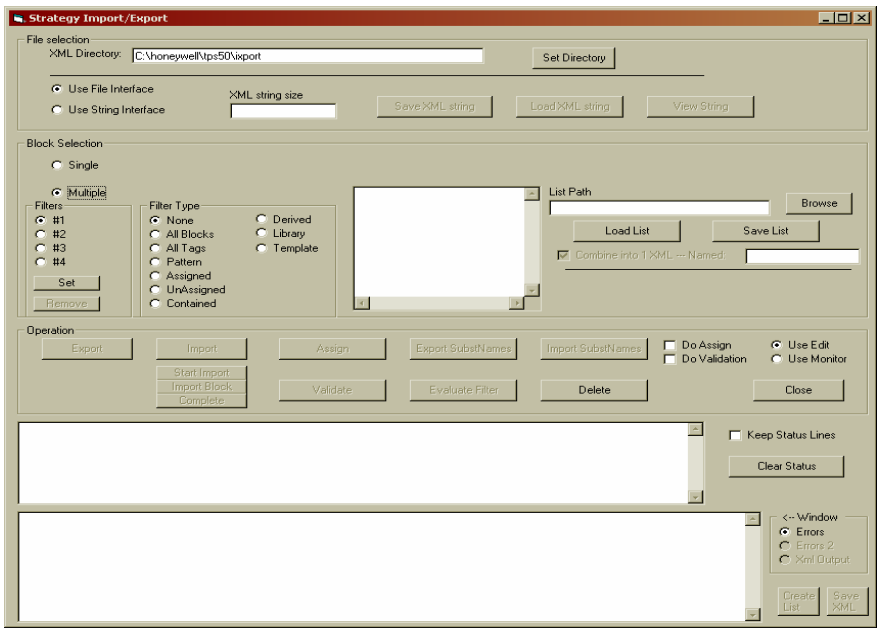
For instances, the Import/Export applies to the selected contained CM(s)/SCM(s). For User Template, Import/Export applied to both the selected CM(s) and its contained CM(s)/SCM(s) if any. Because the containment relationship lives on the contained CM/SCM, if the container CM is imported after the contained CM/SCM is imported, the containment relationship is not established.

Projected Parameter

For instances, the Import/Export applies to the selected contained CM(s)/SCM(s). For User Template, Import/Export is applied to both the selected CM(s) and its contained CM(s)/SCM(s) if any. Because the containment relationship lives on the contained CM/SCM, if the container CM is imported after the contained CM/SCM is imported, the containment relationship is not established.

If a new parameter is added to a function block during development and the name conflicts with a projected parameter name defined prior to this addition in a previous version of the Experion software, the new developer defined parameter will be given precedence. The projected parameter is renamed and all references of this projected parameter are updated.

Strategy Import/Export dialog box



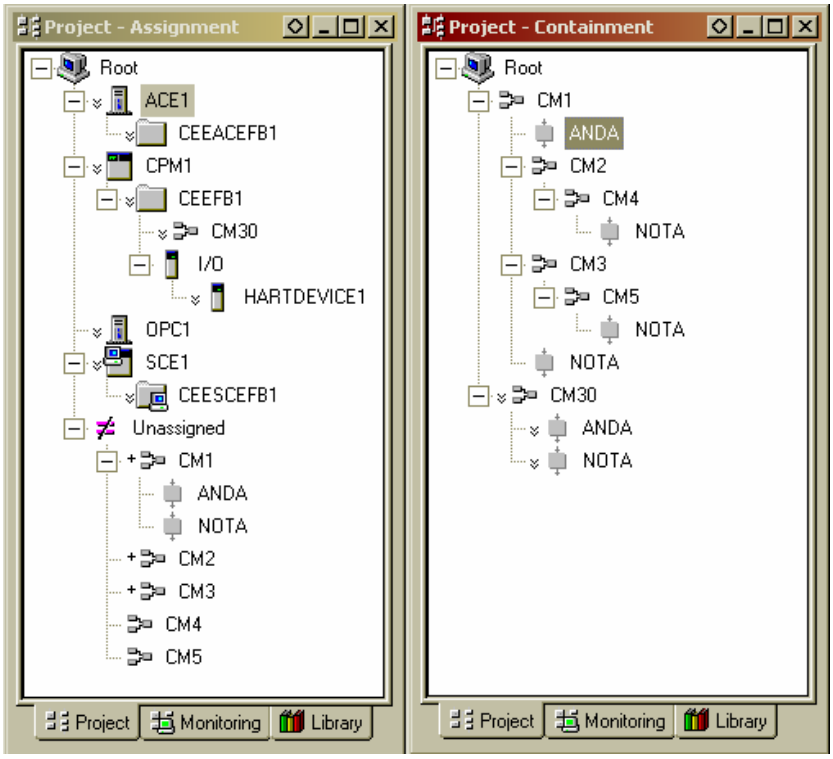
Strategy Import Export dialog box

When Multiple is chosen in the Block Selection list, all Filter Types are displayed.

To set a Filter Type, select the Evaluate Filter button. Then select Xml Output from the Window selection list to display the results of the Filter in the bottom window.

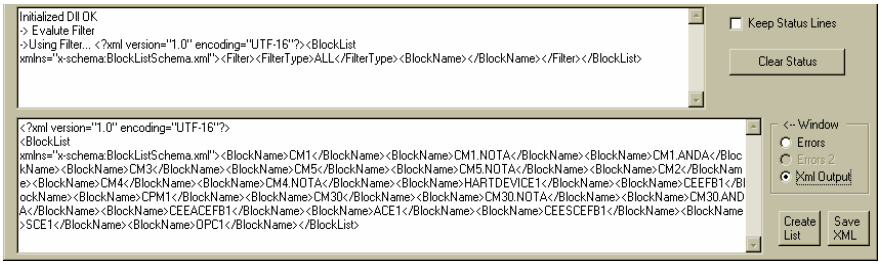
Case	Validation
Filter Type = All Blocks	Displays all the tagged, and basic blocks in the database including blocks at all levels in a hierarchy.
Filter Type = All Tags	Displays all the tagged blocks in the database including tagged blocks at all levels in a hierarchy.
Filter Type = Contained, and entering a CM in the list box.	Displays all the blocks contained by the CM.

For instance, if all the blocks in the database as seen using Control Builder are as shown in the first figure below:

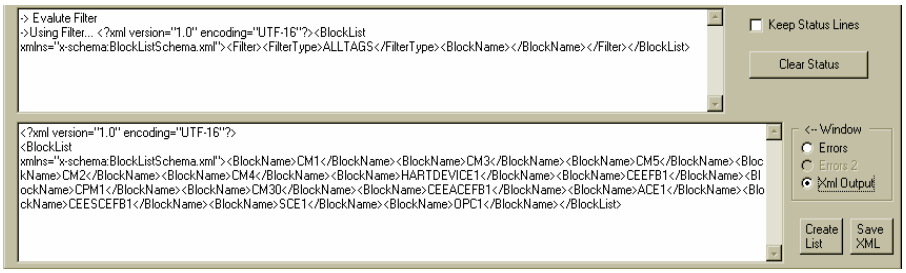


Control Builder view of blocks

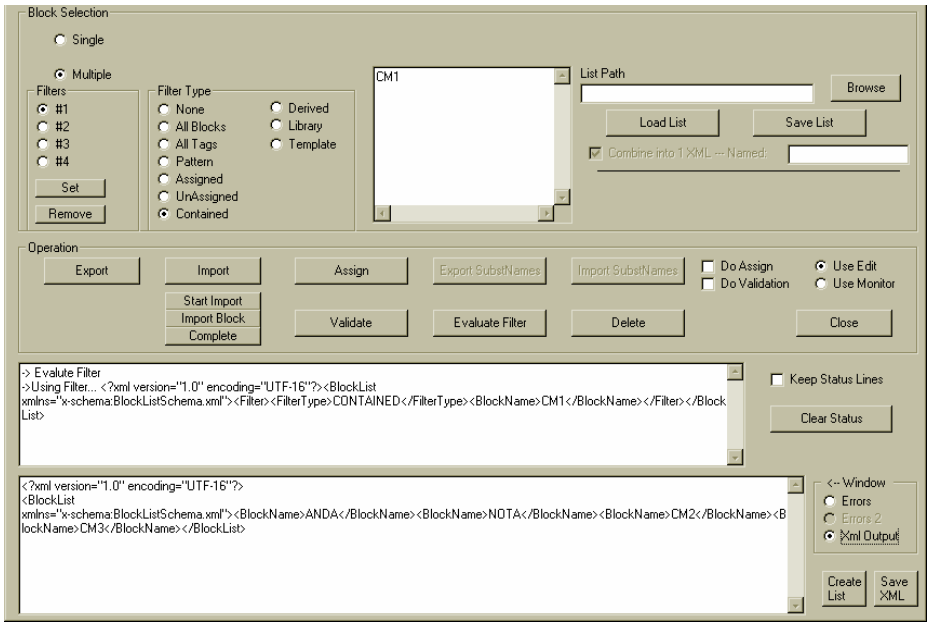
…the XML Output in the dialog box (with various Filter Type settings) should resemble the following figures:



View of XML Output, set to all blocks



View of XML Output, set to all tags



View of XML Output, set to contained

Appendix C - User templates

This section includes information that you can reference for:

Topic	Link
Library tree overview	Click here
Using the library tree	Click here
Project/monitor overview	Click here
Using control drawings	Click here

Library tree overview

Template Libraries are viewed as trees in Control Builder. New Template Libraries can be created and named, and templates can be moved into these libraries.

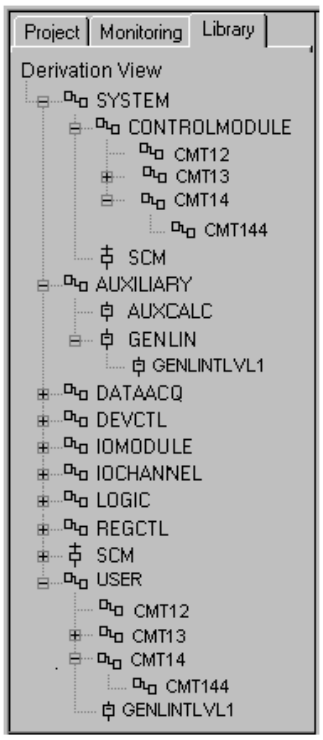
Two views of Template Libraries are available:

- Derivation View - All derived templates are shown as children of the template they are derived from, and in their current user library.
- Containment View - Shows the libraries and their templates in list form. Container templates will show their contained basic blocks.

Derivation View

The following figure shows a tree with Derivation View. Note that some templates appear twice in this view; once under the system template from which they are ultimately derived and once under the user library in which it currently resides. User basic block templates (also known as “standalone” templates) also show up in two places.

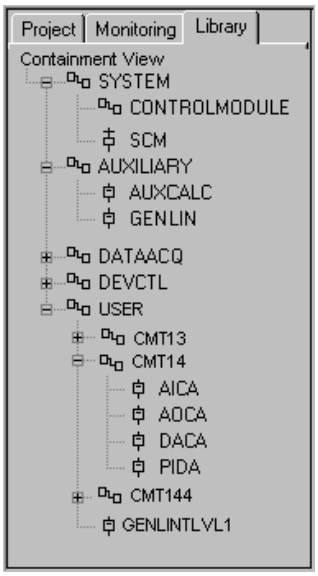
In this example, CMT12, CMT13 and CMT14 are all derived from the system template CONTROLMODULE. CMT144 is derived from CMT14. GENLINTLVL1 is a function block template derived from the GENLIN system template. CMT13, CMT14 and CMT144 appear under both the SYSTEM library and the USER library.



Derivation View

Containment View

The following figure shows a tree with Containment View. Note that templates placed into the user library appear in this view under User. Note that even though CMT144 was derived from CMT14, it appears at the same level in this view.



Using the library tree

This section includes information that you can reference for:

Topic	Link
Switch between Derivation and Containment View	Click here
Configure Module Parameters (user templates only)	Click here
Open control drawing for edit (user templates only), allowed for SCM and CM templates	Click here
Copy (user templates only)	Click here
Move library (user templates only)	Click here
Specialize	Click here
Change parent	Click here
Instantiate (as a tree operation, this applies to container blocks only)	Click here
Promote (user function block templates only)	Click here

Topic	Link
Delete (user templates only)	Click here
Delete library (not functional in this release)	Click here
Identification	Click here
Dependencies	Click here
Import/export template (user templates only)	Click here
Move library (not functional in this release)	Click here

Switch between Derivation and Containment View



TIP: Most Operations supported from the Library Tree are limited to Independent Objects.



TIP: Honeywell System Templates cannot be modified, deleted, or copied. They can only be specialized and instantiated.

- Menu item: View -> Derivation View, View -> Containment View
- Right-click mouse: select Derivation View or Containment View

Configure Module Parameters (user templates only)

- Menu item: Edit-> Module Properties…
- Right-click mouse: select Module Properties…

Open control drawing for edit (user templates only), allowed for SCM and CM templates

- Menu item: File -> Open
- Double-click to open chart.
- Right-click mouse: Open Chart

Copy (user templates only)

Within a user template, Honeywell template library tree, or between library trees a new block can be created with same derivation hierarchy as original block. (Copy differs from Specialization in this regard.)

- Drag and Drop: CTRL + mouse drag/drop
- Copy and Paste: CTRL-C + CTRL-V
- Right-click mouse -> select Copy
- Menu item: Edit -> Copy / Edit -> Paste
- Multiple basic blocks may be selected for copy by clicking the mouse on the objects while holding the Shift key, or drawing a selection rectangle

Move library (user templates only)

Templates may be moved between user libraries or from a system library to any user library. Templates may only be moved from a user library to a system library if the template is ultimately derived from a system template in that library

- Drag and Drop: CTRL + mouse drag/drop

Specialize

Within a user template or Honeywell template library tree: A new block is created that is derived from the original block. (Specialize differs from Copy in this regard.)

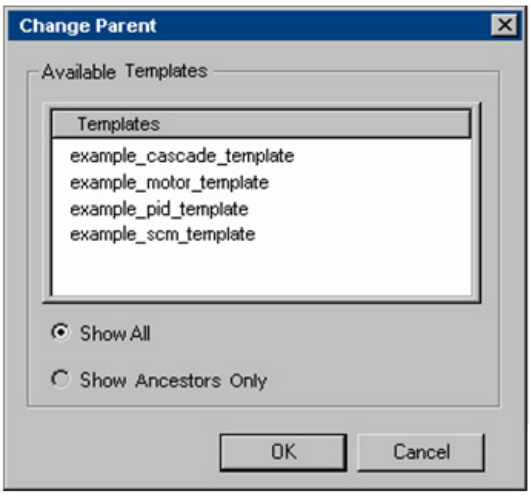
- Right-click mouse -> Make Template
- Menu item: Templates-> Make Template from xxx (where xxx is name of selected block)

Change parent

Within a user template library tree (or Honeywell template library tree): The derivation hierarchy is modified based on the destination of the drag and drop operation.

- Right-click mouse -> Change Parent. A destination block or the library must be selected from the “Change Parent” dialog.
- Menu item: Templates -> Change Parent …
- Use the Change Parent Template Wizard. Refer to

Using the Change Parent Template wizard.



Change Parent dialog box

Instantiate (as a tree operation, this applies to container blocks only)

- Drag and Drop: CTRL + mouse drag/drop
- Right-click mouse -> Instantiate
- Menu item: Templates -> Make Template from xxx (where xxx is name of selected block)

Promote (user function block templates only)

Make an independent template from an embedded function block template.

- Right-click mouse -> Promote
- Menu item: Templates -> Make Template from xxx (where xxx is name of selected block)

Delete (user templates only)

This deletes only the user template. If there are any derived templates and instantiated children - this operation is not allowed.

- CTRL-X
- DEL key on keyboard
- Right-click mouse -> Delete
- Menu item: Edit -> Delete

One or more blocks may be selected. For Delete to complete, the templates must be confirmed by name.

Delete library (not functional in this release.)

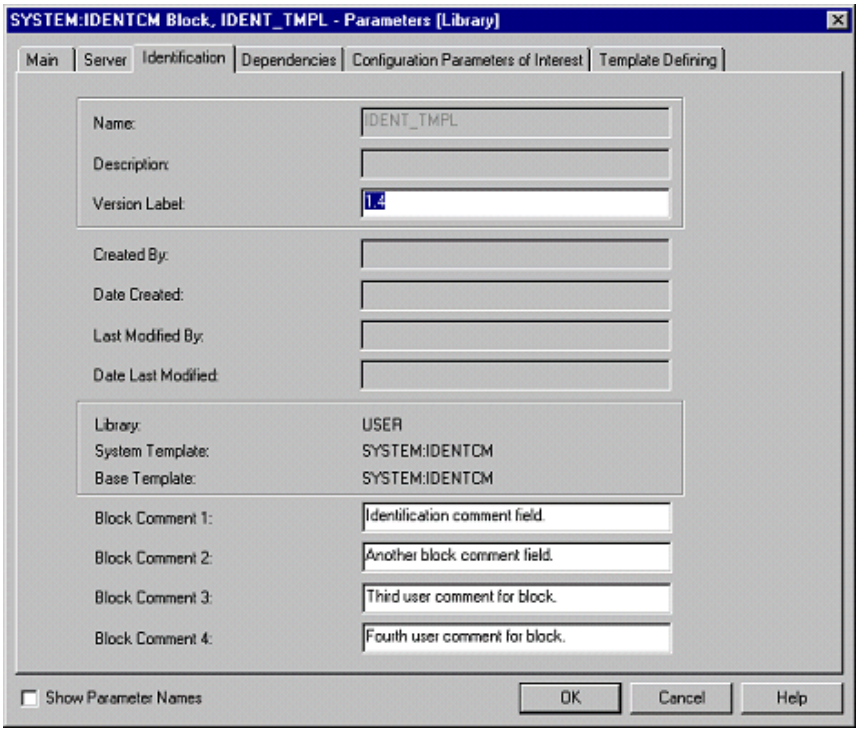
Deletes the library and contained templates only if instances or specialized templates don't exist in other libraries. Honeywell static libraries cannot be deleted. CCL libraries can be deleted if no instances or specialized templates in other libraries exist.

- CTRL-X
- DEL key on keyboard
- Right-click mouse -> Delete
- Menu item: Edit -> Delete

One or more blocks may be selected. For Delete to complete, the libraries must be confirmed by name.

Identification

The Identification tab is located on the configuration parameters form. The following information will be presented on the tab:



The values of template-defining parameters (that is, template-defining attributes defined at *this* level of the derivation hierarchy) are be shown in bold text. Any change of this value will propagate to *all* derived templates and instances.

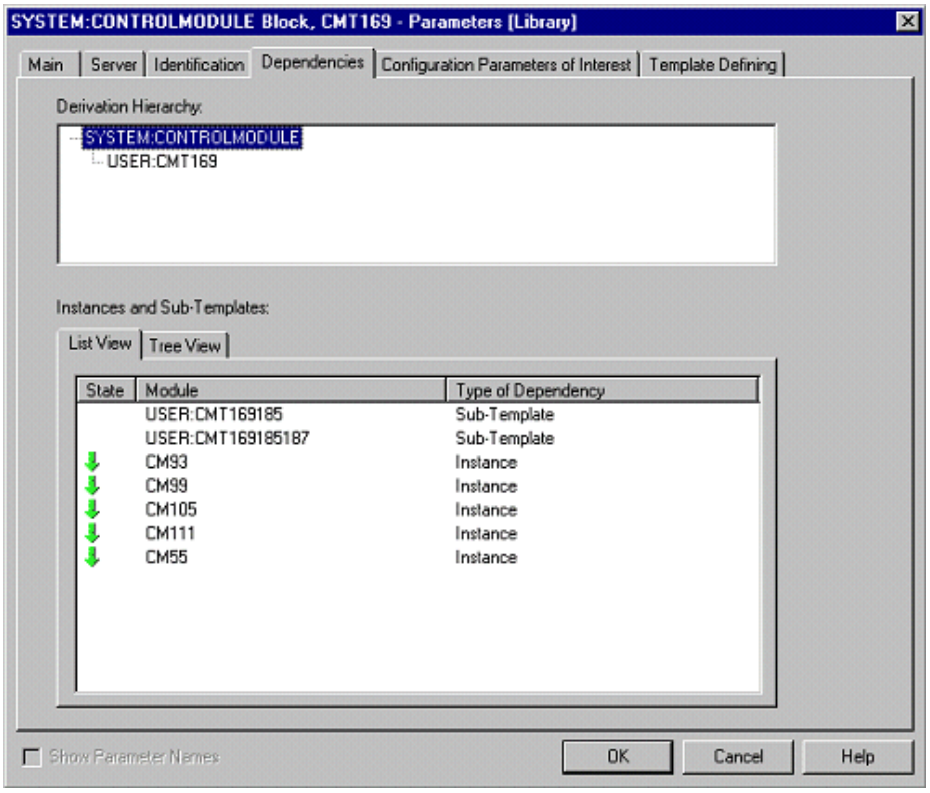
The values of template-defined parameters (that is, those labeled as template-defining at a higher level) are grayed out and in bold text.

Dependencies

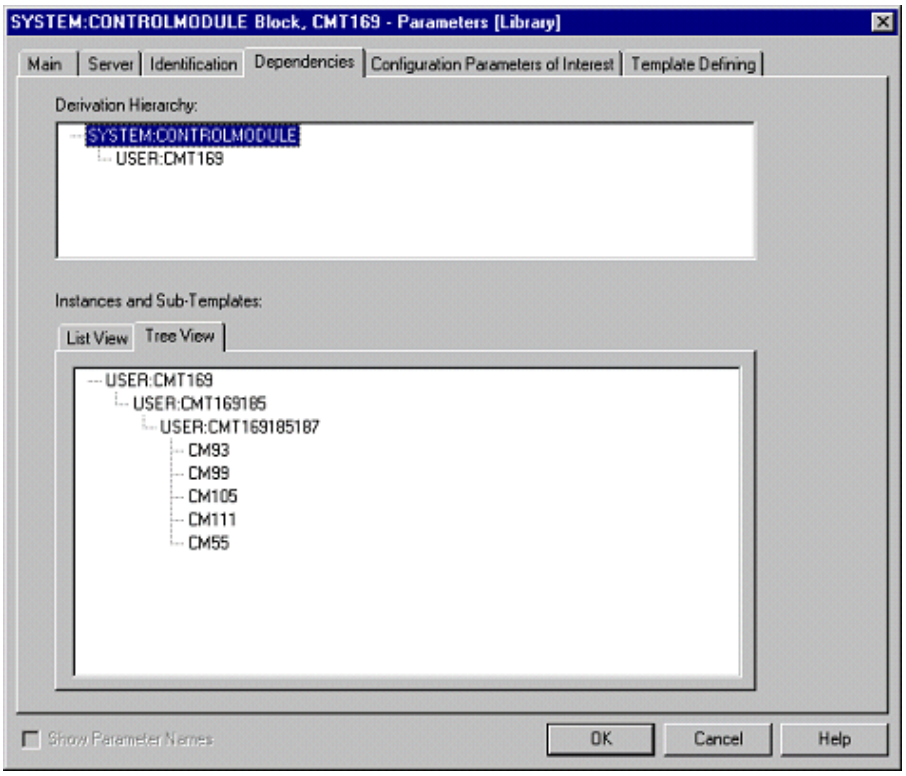
The Dependencies tab is on the configure parameters form. Selecting the Dependencies tab brings up the following dialog, which shows the derivation hierarchy of this template and the sub-templates and instances that are dependent on it.

- The derivation hierarchy (the list of base templates, and base templates of the base templates, ultimately winding back to the original Honeywell system template).
- A list of sub-templates derived from the template.
- A list of instances instantiated from the template and their current life cycle state (loaded, assigned).

Template names are preceded by their library name. Column widths may be adjusted to view longer library and template names. The list may be sorted by dependency type, life cycle state, or the module name. The tree view shows the exact derivation relationship of all dependants.



Dependencies tab, List View



Dependencies tab, Tree View

Import/export template (user templates only)

- Menu item: File -> Import, File -> Export

Move library (not functional in this release.)

Moves library to another tab window.

- Drag and Drop: (CTRL + mouse drag/drop) to another tab window. Confirm the move when prompted.

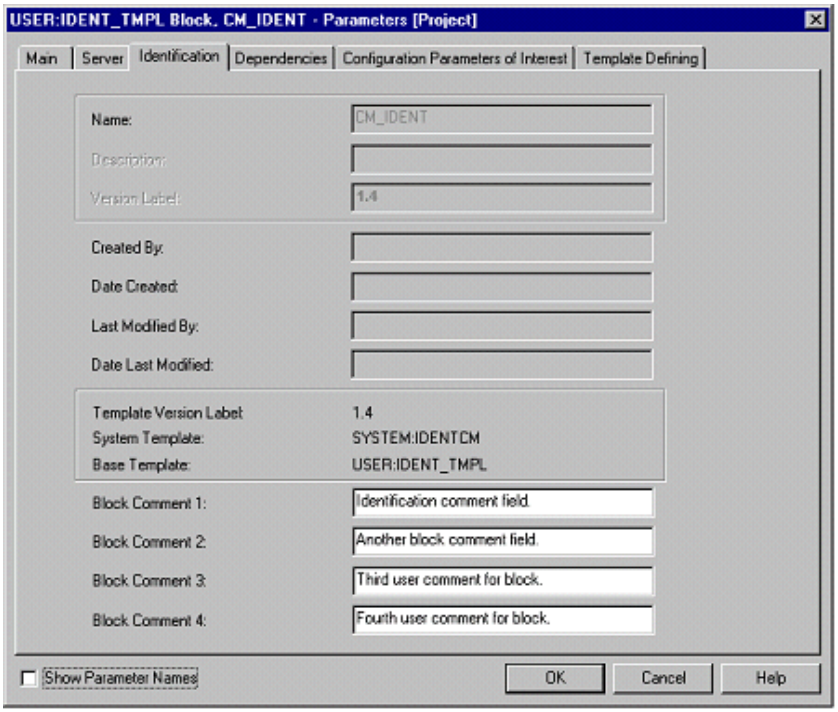
Project/monitor tree overview

The Project/Monitor Tree View allows Instances or strategies to be viewed as trees, showing the assignment relationships in a window in the Control Builder.

If any differences remain between the Project and Monitor databases after an update operation has completed, a yellow delta symbol will appear on the project tree. These differences may be due to a difference in the user template definitions of the strategy in project vs. the definition used at the time the corresponding strategy was loaded to Monitor. If the delta symbol appears, the project must be reloaded to synchronize the project and monitor databases.

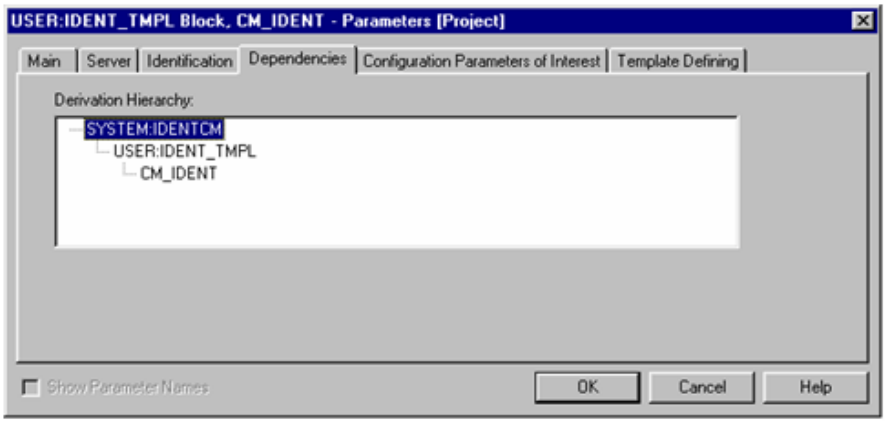
Identification tab

Requesting Configuration Parameters brings up the configuration form, including the Identification tab.



Dependencies tab

Shows the derivation hierarchy for the strategy.



Using control drawings

A control drawing is considered the main work area of Control Builder where control modules are created, function blocks are inserted and connected, and control strategies are initiated.

Opening a control drawing

- Double-click on the template in the Library Tree.
- Select the template in the Library Tree -> select File -> Open
- Select the template in the Library Tree -> right-click mouse -> Open

Copying a control drawing

One or more basic block templates may be copied from one Control Drawing to another. Connections to templates outside of the set of blocks being moved are deleted. Connections within the set of blocks and substituted connections are retained.

- Drag and Drop: (CTRL + mouse drag/drop)
- Copy and Paste: CTRL+C -> CTRL+V
- Right-click mouse -> select Copy
- Menu item: Copy/Paste

Appendix D - Bulk Edit Parameters Utility

This section includes information that you can reference for:

Topic	Link
Overview	Click here
Using Bulk Edit Parameter utility	Click here
Accessing Create Bulk Edit List	Click here
Edit list file examples	Click here
Read Bulk Edit list	Click here

Overview

Bulk Edit Parameters is a utility that allows you to change the value of multiple parameters in a single operation by importing a prepared list of edits.

The Bulk Edit Parameters utility is a feature in Control Builder that allows you to pick tagged object parameters (points) and the parameter values you wish to change. These parameter values can be saved in a spreadsheet file, (or similar) format. The spreadsheet file is then opened in Excel (or a similar application) so that parameter values can be edited as desired. Using the bulk edit parameters utility, the file with its changes is "read" back into the ERDB updating the parameter values.

Editing of simple data type parameters are supported, such as: strings - including enumerations, integer, float, Boolean and arrays of those types. Project parameters are also supported using the edit utility. Binary "Blob" data types re not supported.

Using the Bulk Edit Parameter utility

A "Create Bulk Edit List" function is accessed as an option in the Control Builder "Tools" menu. An interactive dialog allows you to select points and parameters from the database to be included in the bulk edit list. These points and parameter values are identified to be modified and are saved in a spreadsheet file format, (or one of two other file formats). Note that the current ERDB value for the selected parameters is written to the file.

Once the bulk edit list file has been created and saved, you can open the file in the appropriate application, (Excel, Access or Notepad) and then edit or modify the parameter values. Save the bulk edit list file with the changes. You then use the Bulk

Edit Parameter utility to “Read” the bulk edit list file, with the modified values, in Control Builder and update the monitor or ERDB parameter values for each point as required.

The following sections provide more details on the Bulk Edit Parameters utility and give examples on its use.

If your system is licensed for QVCS

If your system configuration includes Qualification and Version Control System, then you must have sufficient privilege to execute QVCS operations, and must be able to check out the objects if modifying the ERDB. If loaded objects are modified and later checked into the ERDB by operations that copy loaded data, (this function will not automatically check in the modified objects to the ERDB), the object could become unloadable, depending on the setting of the fallback qualification state. You should verify the qualification state of any objects modified in this way, and either correct the qualification state or roll back to an earlier, qualified version.

Accessing Create Bulk Edit List

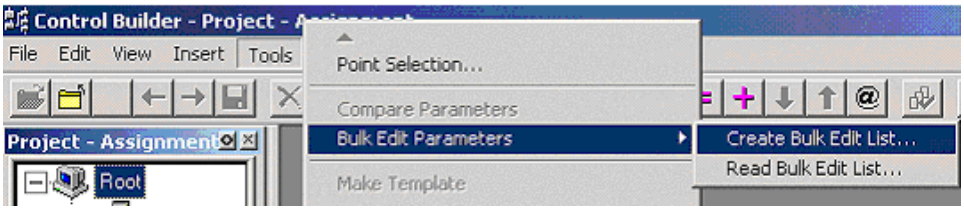
This section includes information that you can reference for:

Topic	Link
Accessing Create Bulk Edit List dialog	Click here
Create Bulk Edit List dialog	Click here
Typical use example	Click here

Accessing Create Bulk Edit List dialog

The Create Bulk Edit List dialog is accessed through the Control Builder “Tools” drop-down menu.

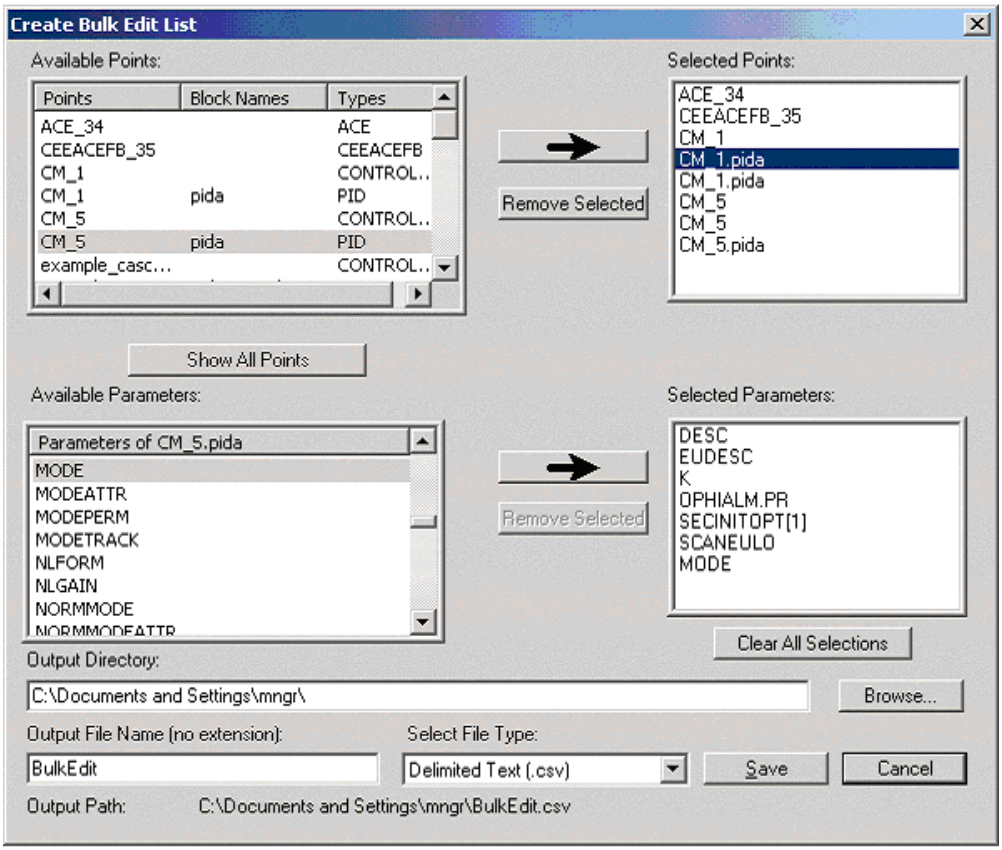
Select: Tools > Bulk Edit Parameters > Create Bulk Edit List …



Launching Create Bulk Edit List dialog

Create Bulk Edit List dialog

The following figure shows the Create Bulk Edit List dialog. The features of this dialog are described in the following table.



Create Bulk Edit List dialog features

Dialog feature	Description
Points :	
Show All Points	This command button is used to populate the “Available Points” selection list with the names of all points in the database. Alternative selection is available using the Browser component.
Available Points	This “point picker” style listview is populated when the the “Show All Points” command button is selected. The right arrow and “Remove Selected” buttons on the right of the listview are enabled when the list is populated, allowing you to copy the currently highlighted point name to the “Selected Points” list. Multiple selection of point names is allowed by using the Shift or Ctrl keys. The point may also be copied by double-clicking on a point name. The data copied will be the “full” point.block name of the selected row.
Selected Points	This list contains the currently selected points to be written to the spreadsheet in “point.block” form. The list is sorted alphabetically as points are added to the list. Selections may be removed at any time by selecting a row and then either double-clicking or using the “Remove Selected” button. This list may alternatively be populated by using the browser to select points with one or more queries.
	Adds the highlighted point(s) in the “Points” list to the “Selected Points” list.
Remove Selected	Removes the highlighted selection from the “Selected Points” list.
Parameters:	
Available Parameters	This list is populated with applicable parameters as points are added to the “Selected Points” list. The “Available Parameters” list functions as a source for adding parameter names to the “Selected Parameters” list. Parameter names are added either by using the right arrow selection button to copy the currently highlighted value (in the “parameters” list), or by double clicking on a parameter name.
Selected Parameters	This list contains the names of all the parameters to be written to the spreadsheet for each point. Parameters may be

Dialog feature	Description
	deselected by using the “Remove Selected” button, or by double clicking on a parameter name in the list. An empty cell for all points indicates that the last appropriate point name was deselected after selecting the parameter(s).
	Adds the highlighted parameters(s) in the “Parameters” list to the “Selected Parameters” list.
Remove Selected	Removes the highlighted selection from the “Selected Parameters” list.

Dialog feature	Description
Other features:	
Clear All Selections	Allows both the Selected Points and Selected Parameters lists to be cleared.
Output Directory	Displays the directory for the storage location of the output file.
Browse…	Displays a directory selection dialog that allows you to choose a different storage location for the output file. The selected directory is displayed in the “output directory” edit box.
Output File Name (No Extension)	Specifies the name of the file to be saved. The file type (and the extension for the name) are selected in the “Select File Type” field.
Output Path	Shows the current full pathname to the output file as assembled from the Output directory, file name, and file type controls.
Select File Type	This drop-down menu allows the selection of the supported output file types and sets the filename extension for the output file.
Save	Creates the output file (spreadsheet, Access database or text), and saves it to the output directory location. The first row of the table will contain the column headers and “Point name”. The remaining columns are set to the selected parameter names. Parameter values for row-column intersections that are invalid will be set to the “empty” (null) value. Excel limitations restrict the total number of different parameters that can be exported to 255, and the total number

	of points (rows) to 65525.
Cancel	Closes the Bulk Edit List dialog without creating or saving an output file.
Help	Activates Knowledge Builder help.

Typical use example

The following procedures outline the steps for building a Bulk Edit list and saving it in a spreadsheet or database file format.

Build a Bulk Edit List

Step	Action	Result/comment
1	Log on to Control Builder as an engineer or manager.	Control Builder application window opens.
2	Select: Tools > Bulk Edit Parameters >Create Bulk Edit List …	The Create Bulk Edit List dialog appears on screen.
3	Click on the “Show All Points” button to populate the “Points Available” list.	The list should fill with all available points in the ERDB. Note that if there are very many points available, the search may take some time so the cursor changes to an hourglass until the list is populated.
4	Select a point in the “Available Points” list.	The “Available Parameters” list will be populated with parameters appropriate to the selected point. (Note: the “Available Points” list permits multiple selections. In the event of multiple selections, the last point in the list controls the “Available Parameters” list).
5	With a point selected, the upper “right arrow” button will become enabled. Pressing this button will copy all selected points to the “Selected Points” list.	The upper “Remove Selected” button becomes enabled when one or more list entries in the “Selected Points” list are selected.
6	Points can be removed from the “Selected Points” list by highlighting the	Point names are removed from the “Selected Points” list.

	point and clicking on the upper “Remove Selected” button.	
7	Multiple rows of a point name may be added to the “Selected Points” list.	This is to allow limited sequencing of parameter changes (e.g. set mode to auto, set sp, set mode to cascade). This can only be accomplished by placing the point in multiple rows of the output spreadsheet, since parameters may only appear once per row. Note that duplicate field names are disallowed by Access and the other database protocol file formats.
8	The “right arrow” and Remove Selected” buttons are used to select or deselect parameters in the “Available Parameters” and “Selected Parameters” list. (Similar to the point selection lists above.) Note that the “Selected Parameters” list does not permit duplicate entries.	Attempts to duplicate an entry will result in the existing entry being highlighted in the “Selected Parameters” list, but not the addition of a duplicate list element.

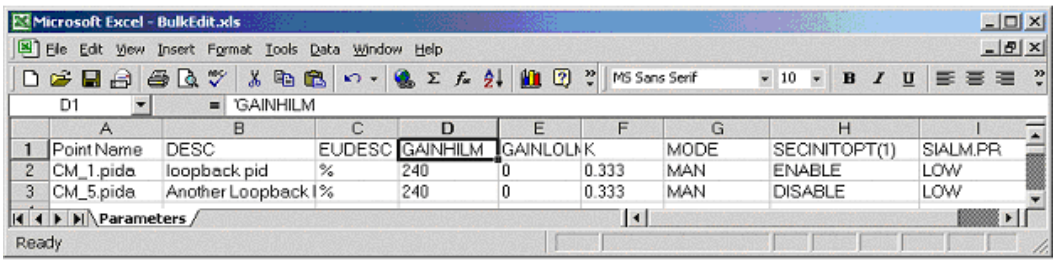
Create Bulk Edit List output file

Step	Action	Result/comment
1	Select a directory for the location of the output file (using the Browse button) or type in the output file pathname.	You can navigate to either a local directory or a mapped network drive. If selected with Browse, the selected directory appears in the “Output Directory” edit control. You can also type in the pathname to the directory into the edit control.
2	Type a name for the output file in the “Output File Name (no extension)” edit control.	It is not necessary to type in a file extension. It is selected using the “Select File Type” dropdown in the next step. Note that the file pathname is indicated in the “Output Path” field at the bottom left of the dialog to indicate the actual output file name and directory that will be used.
3	Select the file type from the “Select File Type” drop down.	This specifies the format of the output file. Supported types are: Excel, Access,

		and delimited text. The file type is appended to the content in the “Output Path” field
4	The “Save” button is enabled ONLY if there is sufficient data entered: Output File Name, File Type and Output Directory, and at least one point and one parameter are in the “selected” lists	Pressing this button will cause the selections in the “Selected Points” and “Selected Parameters” lists to be used in the construction of a properly formatted output file of the chosen file type. The cursor will change to an hourglass while file output creation is in progress.
5	The “Clear All Selections” button may be used to empty both “Selected” lists in the event you wish to start over to create a new bulk edit list.	This is useful to create multiple output files without the need to close and open the Create Bulk Edit List dialog.

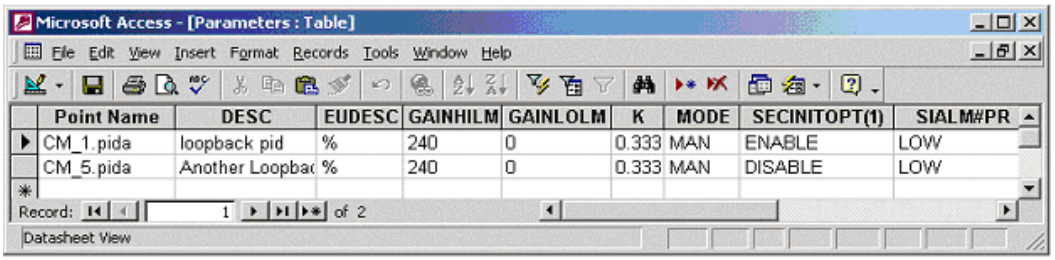
Edit list file examples

The results of creating an edit list file can be seen in the following figures. The first figure shows the arrangement of the points, parameters and their values as displayed in an Excel spreadsheet file format, (.xls extension).



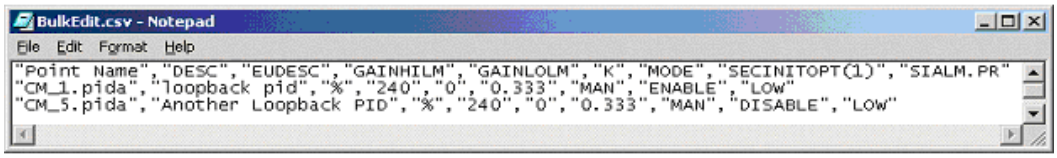
Bulk Edit List as shown in MS Excel file format

The next figure shows an edit list as created using the Microsoft Access database file type format option, (.acs extension).



Bulk Edit List as shown in Microsoft Access file format

The next figure shows an edit list as created using the Delimited Text file type option, (.cvs extension) and viewed using Notepad.



Bulk Edit List in a comma delimited text format

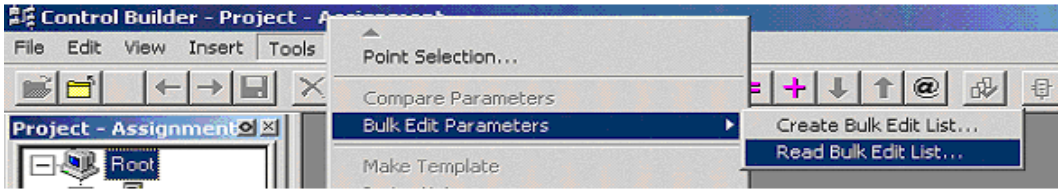
Read Bulk Edit List

The Read Bulk Edit List function is used once you have made changes and saved an edit list file. You use the Read Bulk Edit List to "Read" the file, with the modified values, in Control Builder and update the monitor or project parameter values for each point as required.

Accessing the Read Bulk Edit List dialog

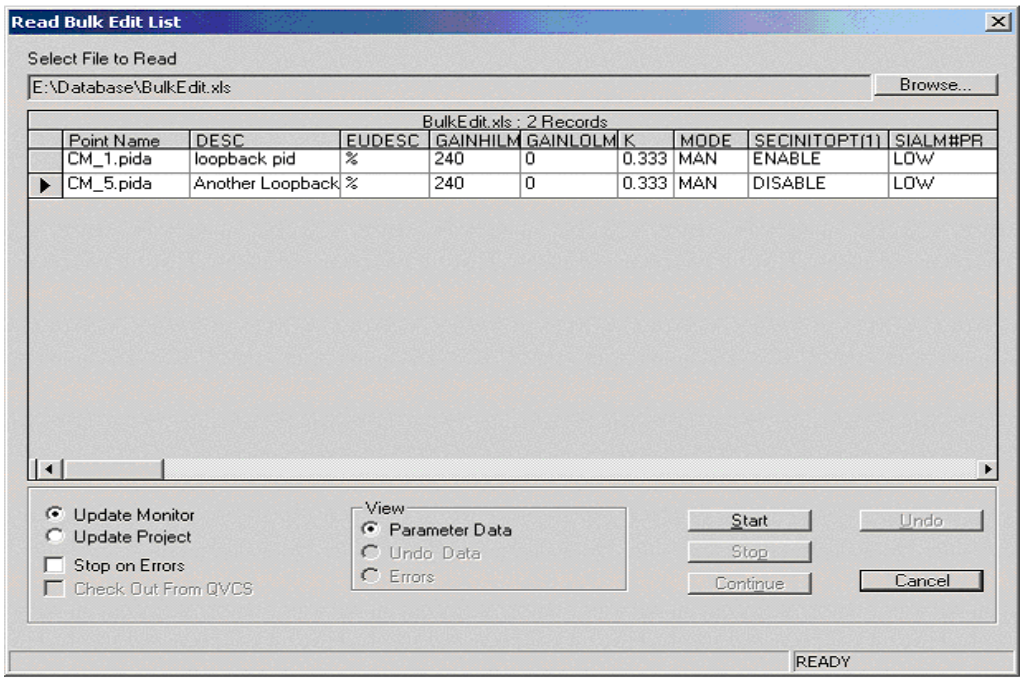
The Read Bulk Edit List dialog is accessed through the Control Builder "Tools" drop-down menu.

Select: Tools > Bulk Edit Parameters > Read Bulk Edit List …



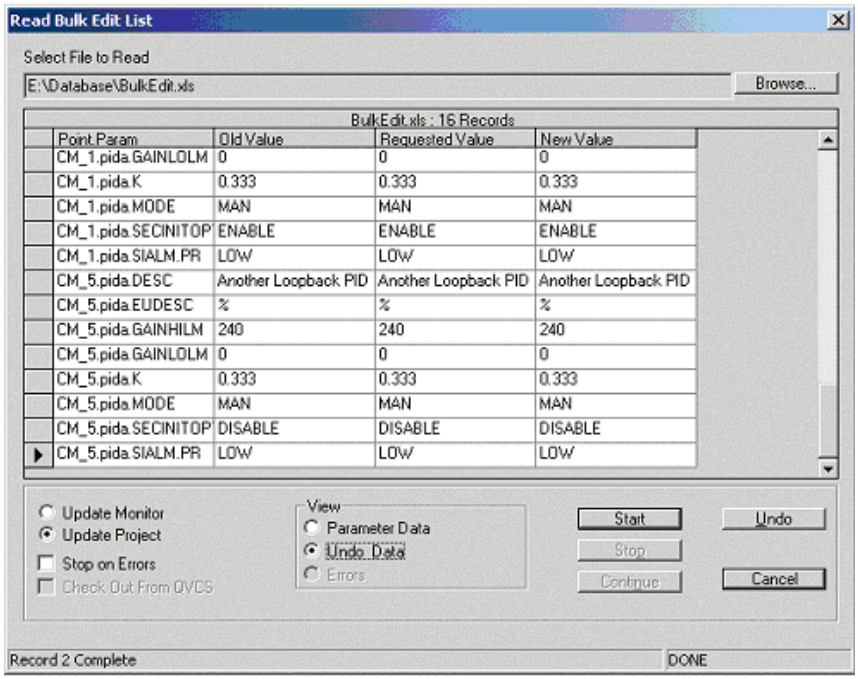
Read Bulk Edit List dialog

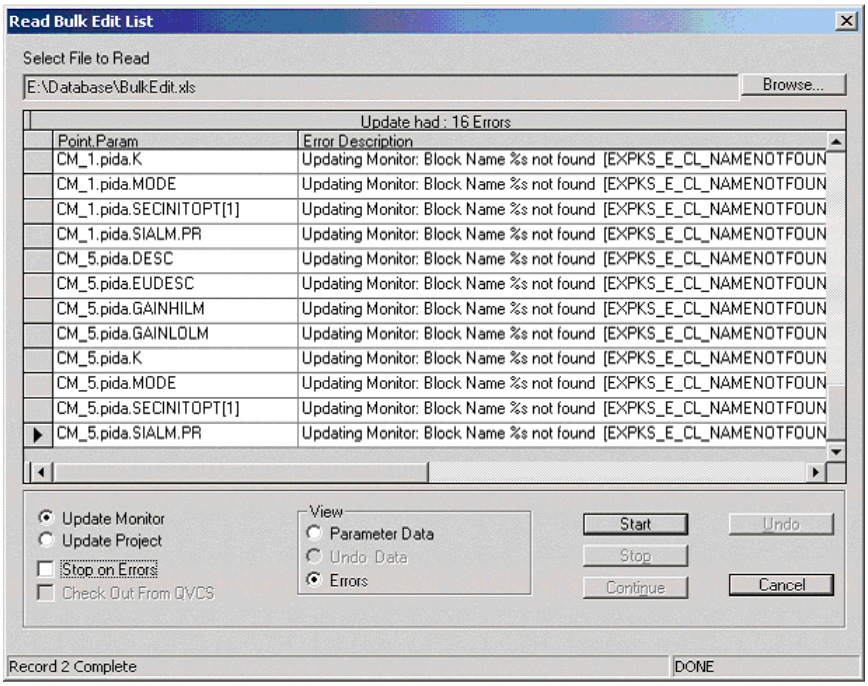
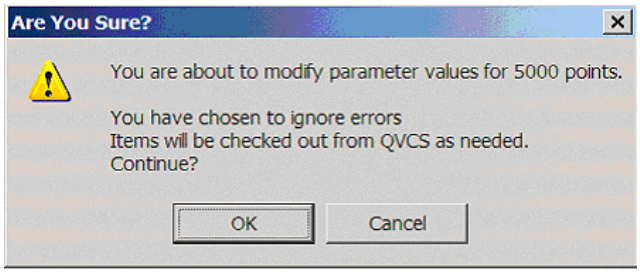
The following figure shows the Read Bulk Edit List dialog. The features of this dialog are described in the following table.



Read Bulk Edit List dialog features

Dialog feature	Description
Select File to Read	Read-only view of the full pathname to the file to read.
Browse	Displays the file selection dialog for selecting the file in the “Select File to Read” control.
Data Grid	Displays the data read from the Excel spreadsheet, or other file formats.
Update Monitor / Update Project	Selects the target environment for the update operation. Update Monitor - Updates parameters loaded to controller function blocks. Note that not all parameters can be updated in active controllers, (for example PVEUHI and PVEULO) Update Project - Updates parameters stored in the Engineering Repository Database.
Stop on Errors	Determines whether errors encountered during the read process cause the update operation to stop or continue until it is completed.
Check Out From QVCS	This is enabled only if QVCS is licensed. It allows the import process to check out any objects not currently checked out during update in “Update Project” mode.
View	Determines the arrangement and type of data to be displayed in the Data Grid area.
Parameter Data	Displays the data in the Bulk Edit List file currently in use. The data is displayed in the same format as the original output file, one point name per row.
Undo Data	Displays the results of the update. It shows one row per point and parameter as shown in the following figure, with: – the original parameter value, – the requested parameter value, – the new parameter value.

Dialog feature	Description
	
Errors	Displays any errors that occurred during the parameter update operation.

Dialog feature	Description
	
Start	Enabled when an appropriate file is selected and read into the grid, it starts the update processing of the list. When selected, a confirmation dialog will be shown with the selected options (Monitor or Process, Stop on Errors, etc.) and a warning that multiple updates will be made to the parameter values in the system. 

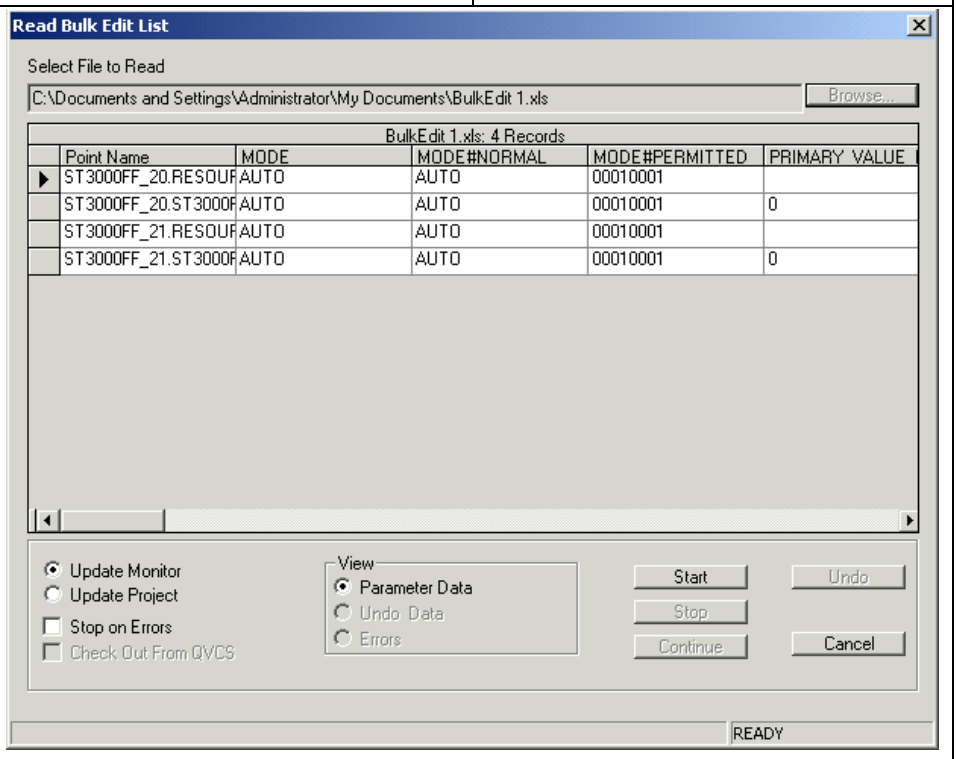
Dialog feature	Description
Stop	Will stop the update process at the completion of the current data grid row.
Continue	Enabled any time the update process has stopped with pending rows remaining in the data grid. Allows the current process to continue.
Undo	Causes the process to restart and resets the values of all points updated so far. The utility resets all values that were changed back to the previous values before the current file read operation.
Cancel	Enabled any time updates are not active, closes the dialog.
Status Bar	Displays the current state of the import process, including the point being processed.
Help	Activates Knowledge Builder help.

Reading a Bulk Edit List (a typical use example)

The following procedure outlines the steps to read an edit file list and update the parameter values of selected points.

Step	Action	Result/comment
1	Log on to Control builder as an engineer or manager.	Control Builder application window opens.
2	Select: Tools > Bulk Edit Parameters > Read Bulk Edit List …	The “Read Bulk Edit List” Dialog appears on screen. Initially the dialog shows an empty grid and only the “Browse…” and “Cancel” command buttons enabled. “View” options are all disabled. The right pane of the status bar displays a “NOT READY” status.
3	Select an edit list file to read. Press the “Browse” button to display a file selection dialog and navigate to the desired file directory.	The file selection dialog shows only file types that are supported by the utility: Excel Spread Sheet, Microsoft Access Database, and Delimited Text (.csv). Attempting to open a spreadsheet that

Step	Action	Result/comment
		does not contain the expected format will result in an error dialog, and the status will remain “NOT READY”.
4	When a valid file is chosen, the selection dialog closes with the “OK” button	The selected file is read into the grid in the Read Bulk Edit List dialog, and the Start button is enabled. The status bar indication changes to “READY”. The Grid display updates to show the content of only the “Parameters” spreadsheet (sheets containing any other names are ignored). The Grid caption displays the file name and the number of records (points) in the sheet. The “Parameter data” view becomes enabled and selected, and the “Start” command button is enabled. Note that the grid data is read only (to prevent inadvertent discrepancies between the list used and the actual update values), but it will allow you to inspect the data in the edit list file for the update session.

Step	Action	Result/comment
		
5	Choose to update either of the two environments: • Update Monitor values or • Update Project values	Option selectors are located at lower left of dialog.
6	Choose either processing behaviors: • Check the “Stop on Errors” box to stop processing at the first error, or • Uncheck the “Stop on Errors” box to continue processing to the end of the list before stopping.	If errors occur during the update operation, the “Errors” control will be enabled.
7	If “Full Qualification and Versioning” are licensed, the “Check Out From QVCS” checkbox is enabled.	If the “Check Out From QVCS” check box is checked, project parameters are automatically checked out so that they

Step	Action	Result/comment
	Click on check box to enable this feature.	may be updated during the update operation. “Check Out From QVCS” check box is ignored if “Update Monitor” mode control is selected.
8	Press the “Start” button to initiate the update operation on a background priority.	The “Stop” button becomes enabled, and the “Start” button becomes disabled. As updated values are processed, the completed row is displayed in the left status bar pane, and the row selector indication in the grid advances. If the “Stop” button is pressed, or if “Stop on Errors” is checked and an error occurs, then the update operation pauses at the end of the current row. The “Undo,” “Continue” and “Cancel” buttons are enabled. The “Undo” button is enabled only if any updates have succeeded. (It is possible that all attempted updates have resulted in errors.) If any updates have been processed, the “Undo Data” view option is enabled. If any errors have occurred the “Errors” View Option is also enabled
9	When the update operation has completed (all spreadsheet rows have been processed), the “Start” button is enabled and the “Continue” button is disabled	
10	Selecting the “Undo Data” View option will display the update results in the grid.	This is a list by Point.Parameter of the old, requested and new values.
11	If the “Undo” button is pressed, the undo list is processed, starting with the last row and working backwards to the first.	

Appendix E - User Defined Symbols

This section includes information that you can reference for:

Topic	Link
Introduction	Click here
Viewing symbol library	Click here
Creating a user defined symbol	Click here
Adding symbol to library	Click here
Deleting symbol from user library	Click here
Reloading user defined symbol	Click here
Exporting symbol	Click here
Assigning symbol to block	Click here
User defined template considerations	Click here
QVCS and UDS considerations	Click here
Defining automatic pin and UDS	Click here
Defining tree displays and UDS	Click here
Specifying chart header and footer layouts	Click here
Printing a chart	Click here
Conventions	

Introduction

User Defined Symbols (UDS) may be created in Control Builder as an alternative representation of the standard Control Builder Symbols. The following figure shows the default or standard symbol used to represent an AND logic block in Control Builder.



Standard Control Builder symbol

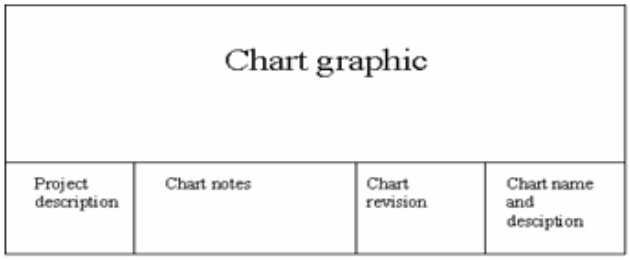
Depending on the application, the representation that is desired may be different. For example, the Power Generation group uses SAMA standard symbols when documenting control strategies. The following figure shows an alternate UDS that could be used to represent an AND logic block in Control Builder.



Alternate Control Builder symbol

The ability to define custom symbols provides increased flexibility, and where documentation is required to follow specific guidelines for symbols, significant cost savings can be achieved.

In addition to the custom symbol support, flexibility in the formatting and content of headers and footers of chart printouts is available. These help provide a consistent user documentation style and content. An example for the type of layout for a printed chart is shown below.



Example chart layout

Viewing symbol library

A system library that contains a set of standard SAMA symbols is provided with the Symbol Library that is included with Control Builder. You cannot change these symbols, but you can assign a given symbol to a selected function block. You can also add new symbols to the Symbol Library or delete user symbols from the library as well as reload an image from the source file or export an image to another file location.

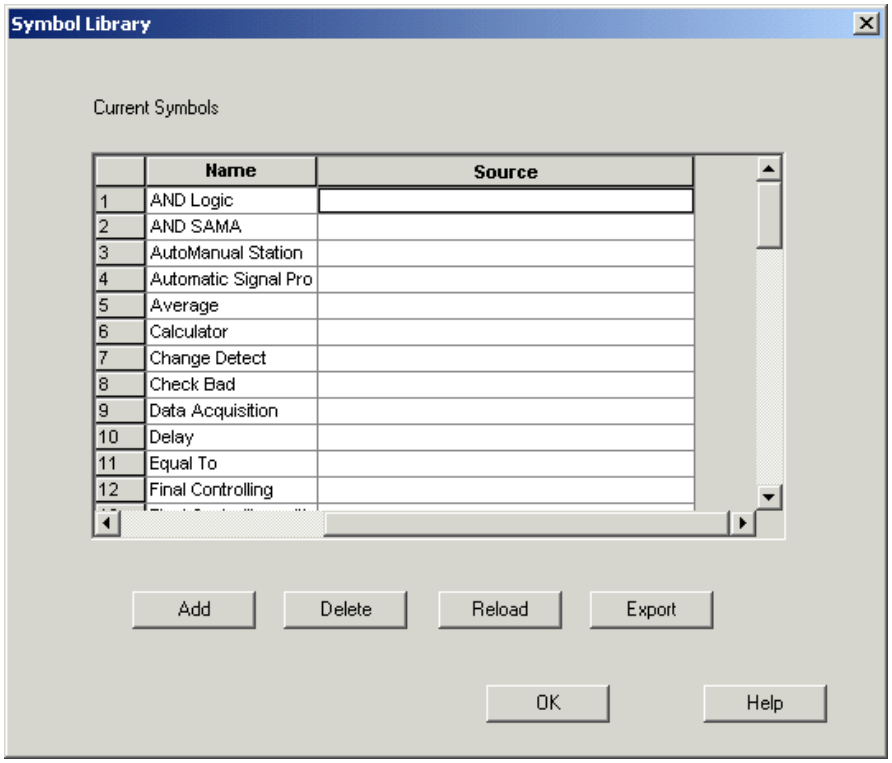
Prerequisites:

- You have installed Experion R200 or greater software and have launched the Control Builder Application.
- You have logged on with a security level of Manager. Access to the Symbol Library is restricted to inhibit unauthorized modifications.

Considerations:

- You are familiar with interacting with applications in a Windows environment.
- You can access the **Symbol Library** from the **Tools** menu or right-click in the **Project**, **Monitoring** or **Library** tab.

Step	Action
1	Right-click in **Project** tab and select **Symbol Library**.
2	View **Symbol Library** dialog on top of Control Builder.

Step	Action
	
3	See other procedures in this section to Add, Delete, Reload, or Export a symbol.
4	This completes the procedure. Go to another procedure in this section or click the **OK** button to close the dialog.

Creating a user defined symbol

This section includes information that you can reference for:

Topic	Link
Supported graphics formats	Click here
Image considerations	Click here
Examples	Click here

Supported graphics formats

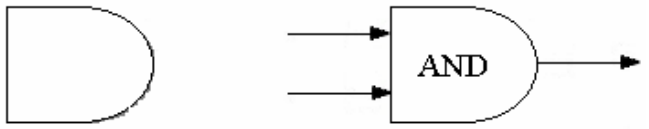
You can create User Defined Symbols using any standard graphics program, although only the Windows bitmap file format (.bmp) is supported in Experion. Experion does not provide support for creating the image files. There are many graphic programs available, ranging from the fairly basic capabilities of the Windows Paint program, to sophisticated applications such as Adobe Illustrator, Corel Draw, Corel Paint, etc. Use the program that you are most familiar to create the symbol and then save the graphic as a bitmap file (.bmp).

Image considerations

- When creating a symbol image, be sure you consider how the symbol will appear on a display, as well as in print.
- A symbol should have a distinct boundary so that any connections to other symbols can be easily seen.
- Typically text should not be included as part of the image as that can be generated programmatically when the symbol is displayed or printed.
- Consider the size of the image, as very large or small images may not scale cleanly or may not fit well with other symbols. Also, the size is important, if faceplate parameters and pins will be defined for the symbol. Note that the actual size of the symbol is determined by the size of the image, not the location of the symbol borders within the image. Typically, the image should be sized such that the symbol borders are quite close to the height and width of the image.
- The careful use of color is also a factor, as too much color can be confusing.

Examples

As an example, consider a symbol for the logical AND block.



AND block example

The very basic shape shown on the left is a good example for a user defined symbol. It has a clearly defined border and no text.

The shape shown on the right is not as good because it reduces flexibility of the number of connection pins, and the "AND" text cannot be changed with the actual name assigned to the control block.

Using the symbol on the left, and by defining its attributes, the following could be the resulting display:



User symbol with displayed attributes

An example of a chart displayed with standard Experion symbols, and the same chart displayed with a mix of UDS (in this example, based on SAMA) and standard Experion symbols is shown in the following figures.

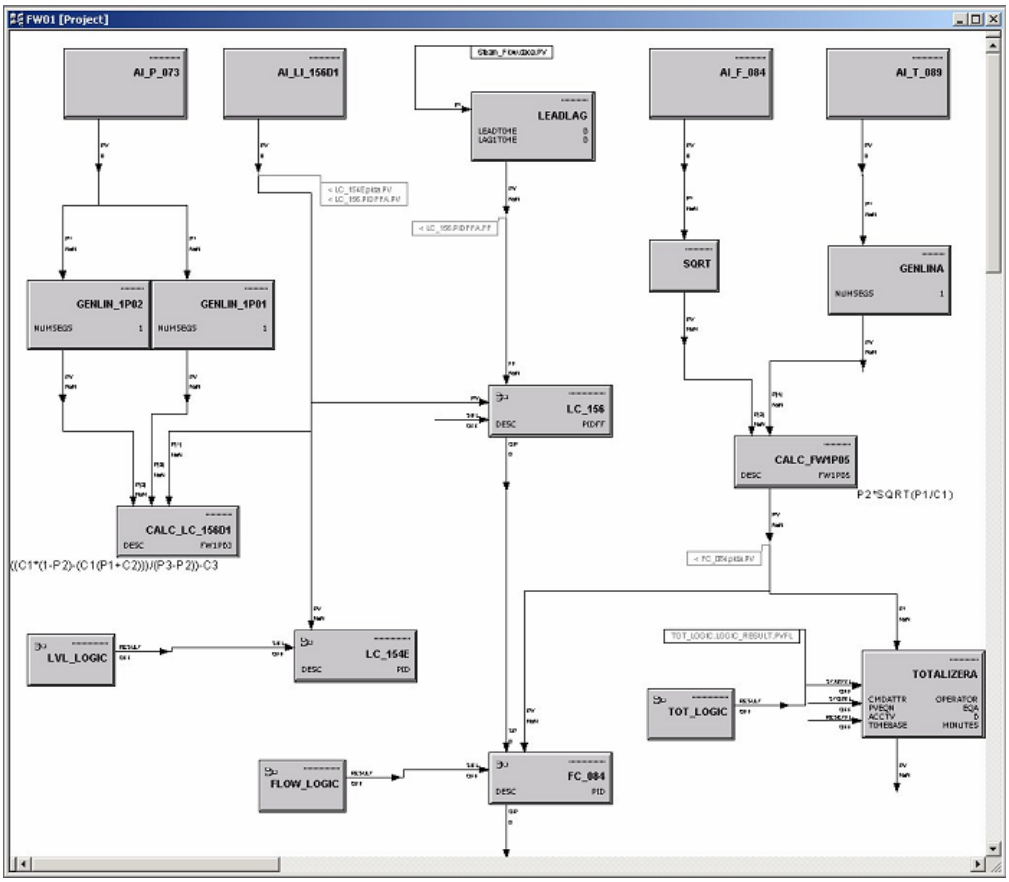


Chart with standard Experion symbols

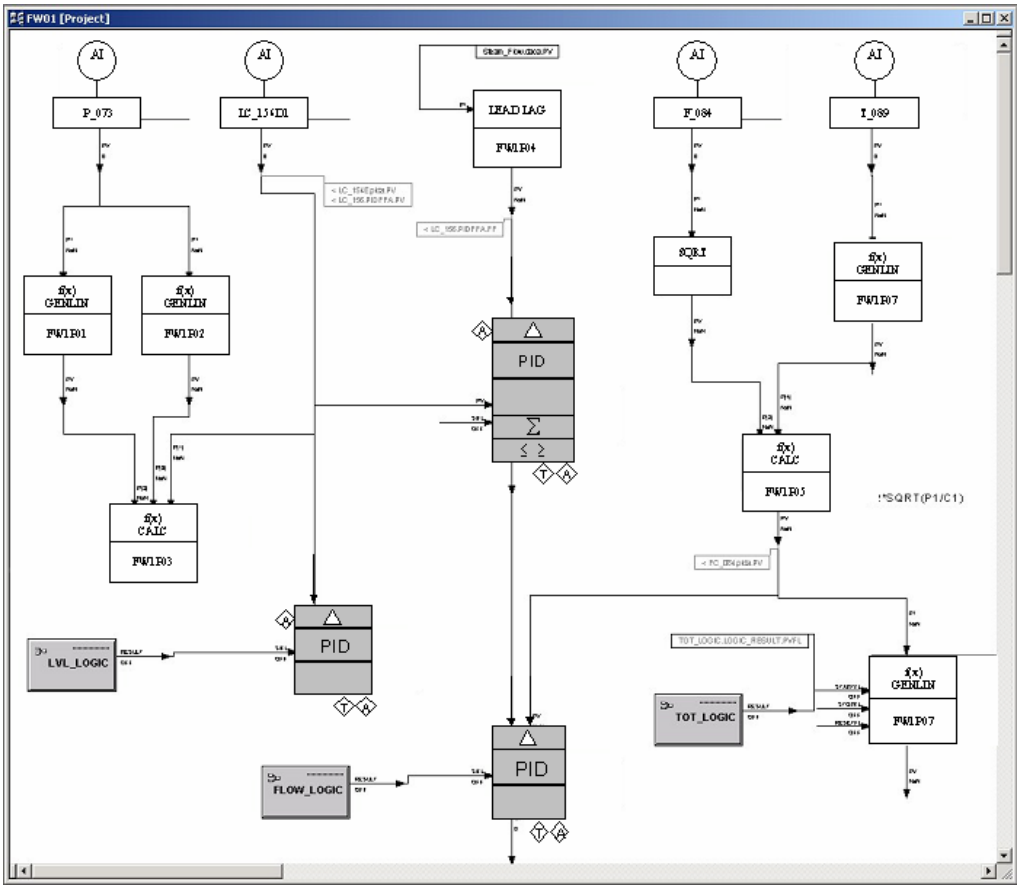


Chart with user defined symbols

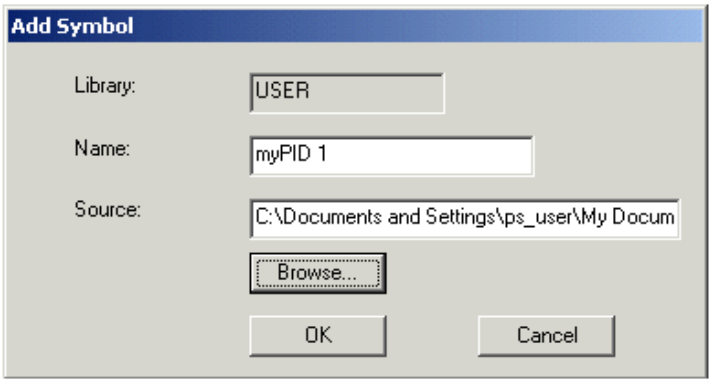
Adding symbol to library

Prerequisites

You have completed the previous *Viewing symbol library* procedure.

Considerations

The symbol to be added is in a compatible graphics format as noted in the table above.

Step	Action
1	In **Symbol Library** dialog, click the **Add** button.
2	In **Add Symbol** dialog Name box, key in desired name for UDS consisting of up to 49 characters.
3	In **Source** box, key in directory location were UDS symbol graphic file is stored or click the **Browse** button to use the **Select Image File** dialog to navigate to the location. The following illustration is for example purposes only.
	
4	Click the **OK** button to add the named image to the User Library within the Symbol Library.
5	In the **Symbol Library** dialog, scoll the **Current Symbols** list to find the image just added in the USER Library.
6	Repeat this procedure on add other symbols as required.
7	This completes the procedure. Close the **Symbol Library** dialog or go to another procedure in this section.

Deleting symbol from user library

Prerequisites

You have completed the previous *Viewing symbol library* procedure.

Considerations

- The symbol to be deleted resides in the User Library within the Symbol Library.
- The symbol to be deleted is not referenced by a block in the control strategy. You can not delete an image if it has been assigned to a block in Control Builder.

Step	Action
1	In **Symbol Library** dialog, scoll the **Current Symbols** list to find the image you want to delete from the USER Library.
2	Click the name of the image to be deleted.
3	Click the **Delete** button. Click the **Yes** button to acknowlege the action.
4	Click the **OK** button to confirm the action.
5	Check that image is deleted from the **Current Symbols** list.
6	This completes the procedure. Close the **Symbol Library** dialog or go to another procedure in this section.

Reloading user defined symbol

Prerequisites

You have completed the previous *Viewing symbol library* procedure.

Considerations

- The symbol to be reloaded resides in the User Library within the Symbol Library.
- Once you add a symbol to the User Library within the Symbol Library, the original graphic file is not used by Experion. This means the original file may be moved, renamed, or deleted, and there is no impact on the ability to display the symbol. This also means that any changes made to the original file will not be reflected in the displayed symbol unless the Reload function is invoked.

Step	Action
1	In **Symbol Library** dialog, scoll the **Current Symbols** list to find the image you want to reload in the USER Library.
2	Click the name of the image to be reloaded.
3	Click the **Reload** button.
4	Click the **OK** button to acknowledge the image has been reloaded from the source listed.
5	This completes the procedure. Close the **Symbol Library** dialog or go to another procedure in this section.

Exporting symbol

Prerequisites

You have completed the previous *Viewing symbol library* procedure.

Considerations

The exported symbol is the version that resides in the Symbol Library on the ERDB and not the one that resides in the source file.

Step	Action
1	In **Symbol** Library dialog, scoll the **Current Symbols** list to find the image you want to export.
2	Click the name of the image to be exported.
3	Click the **Export** button.
4	In **Save As** dialog, navigate to file location where exported image is to be stored. Click the Save button.
5	Click the **OK** button to acknowledge the location where the image has been stored.
6	This completes the procedure. Close the **Symbol Library** dialog or go to another procedure in this section.

Assigning symbol to block

You assign a user defined symbol to a block through the **Block Preferences** tab on the block's configuration form.

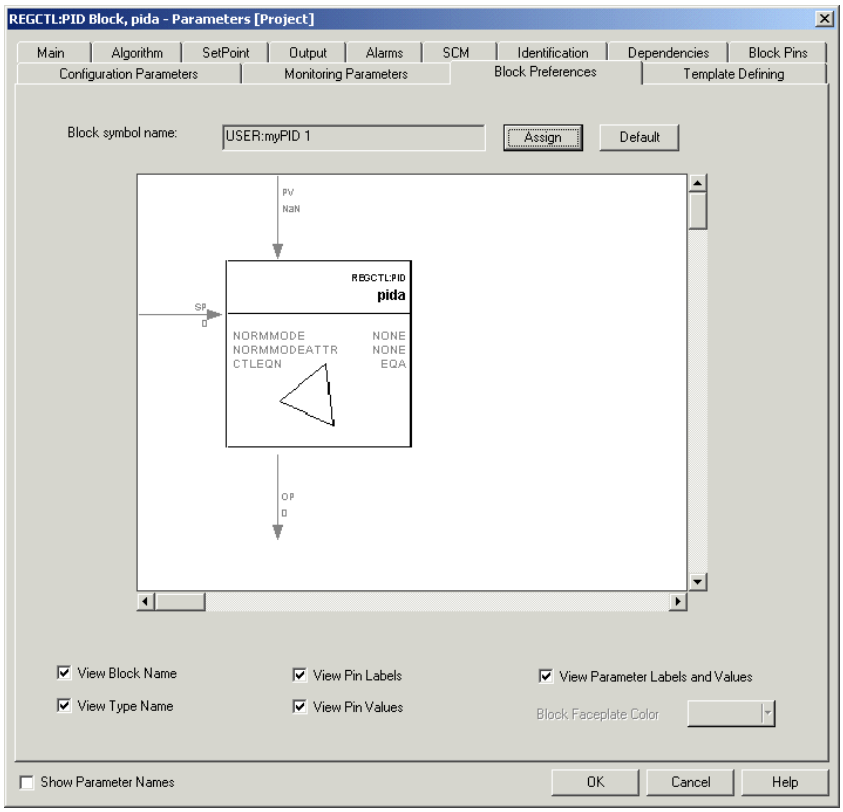
Prerequisites

You have installed Experion R200 or greater software and have launched the Control Builder Application.

Considerations

- You have added any symbols you want to use to the User Library within the Symbol Library. See the previous *Adding symbol to library* procedure for details.
- You have created a Control Module with contained blocks.
- When changing the block-associated symbol, all defined attributes for that block such as faceplate parameters and pins remain unchanged, but may be repositioned.
- The symbol name for an assigned user defined symbol appears in the **Block symbol name** box on the **Block Preferences** tab and cannot be modified.
- You can click and drag the block name and type name as a linked pair to reposition them on assigned symbol on the **Block Preferences** tab.
- You can click and drag input and output pins to reposition them on assigned symbol on the **Block Pins** tab.
- You can click and drag parameter listings to reposition them on the faceplate of assigned symbol on the **Configuration Parameters** and **Monitoring Parameters** tabs.

Step	Action
1	In **Project** tab, click the + sign for the Control Module icon that contains the function block you want to assign a symbol to. For example, click the + sign for the **example_pid** control module icon to expose its blocks.
2	Right-click the icon for desired block and select **BlockProperties**. For example, right-click the **pida** block icon contained in the **example_pid** control module.
3	Click the **Block Preference** tab.
4	Click the **Assign** button.

Step	Action
5	On **Assign User Symbol** dialog, use vertical scroll bar to view contents of symbol library, click name of desired symbol, click the **OK** button to select the symbol and close the dialog.
6	The assigned symbol is now displayed in the view box. Use configuration functions on the tab to change the appearance of the block as desired. Click the **Help** button for information about the functions. The following illustration is for example purposes only.
	
7	Configure **Block Pins**, **Configuration Parameters**, and **Monitoring Parameters** as you would for any block. See *Using the Parameters Configuration form* for more information.
8	Do you want to revert to the standard Control Builder symbol for this block?

Step	Action
	• If the answer is **Yes**, click the **Default** button to replace the assigned user symbol with the standard one. Click the **Yes** button to acknowledge the action. • If the answer is **No**, go to the next Step.
9	On the configuration form, click the **OK** button to save the changes and close the form.
10	This completes the procedure.

User defined template considerations

If you wish to replace the default symbol for all instances of a block, you can assign a UDS to a user defined template for the block. That template (instead of the system template) is then used anytime that block type is instantiated.

When you configure a user template, be aware that any block preferences, block pins, configuration parameters, and monitoring parameters that are defined are treated as template defining attributes, and cannot be altered when the template is instantiated. For example, if a parameter, say IN(1), is defined as an input block pin on the template, it will appear on all instances, and it cannot be deleted or moved for an instance. Other pins can be added and moved, but that pin will remain unless it is deleted or moved on the user template.

The template defining characteristic is also extended to the display and placement of the block and parent template name, and also the symbol assignment itself. That is, the symbol assigned to an instance of a block cannot be changed for that block, except by changing the assignment on the user template.

QVCS and UDS considerations

As for any parameter for an object, the changing of the symbol assignment, is subject to the normal restrictions imposed by the Qualification and Version Control System (QVCS). That is, the object must be checked out before a change in the assignment (or in any of its attributes such as pins, faceplate parameters, etc.) can be made.

The actual symbol (in the Symbol Library) is not subject to QVCS controls. Thus, changes may be made to the appearance of a symbol without having to check out all of the objects that reference that symbol. Because the symbol that is displayed does not affect the functionality of the block, there is no loss of qualification state criteria.

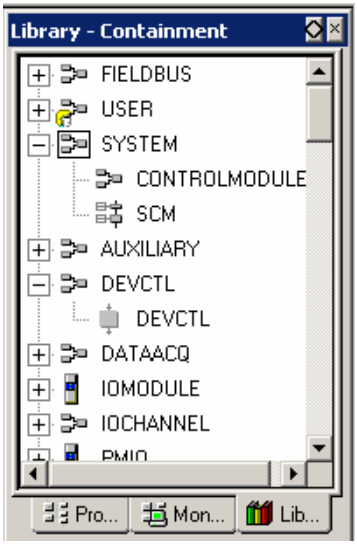
Defining automatic pin and UDS

A feature that is supported by Control Builder is the automatic generation of pins. This occurs when a parameter connection is made to a block that does not have the parameter exposed as a pin. In this case, Control Builder will automatically add that parameter as a pin. This pin will be added to the end of the list of pins, and becomes part of the attribute definition for that block. A side effect of this behavior is that the pin that is automatically generated may not appear "connected" to the graphic for a UDS. If this occurs, it is a simple matter to configure the placement of the pin using the methods described for configuring pins.

Defining tree displays and UDS

For navigation of the **Project**, **Monitoring** and **Library** tabs, their contents are displayed in tree controls that display an icon and name for each object. A sample of a **Library** tab with some of the icons is shown in the Figure below.

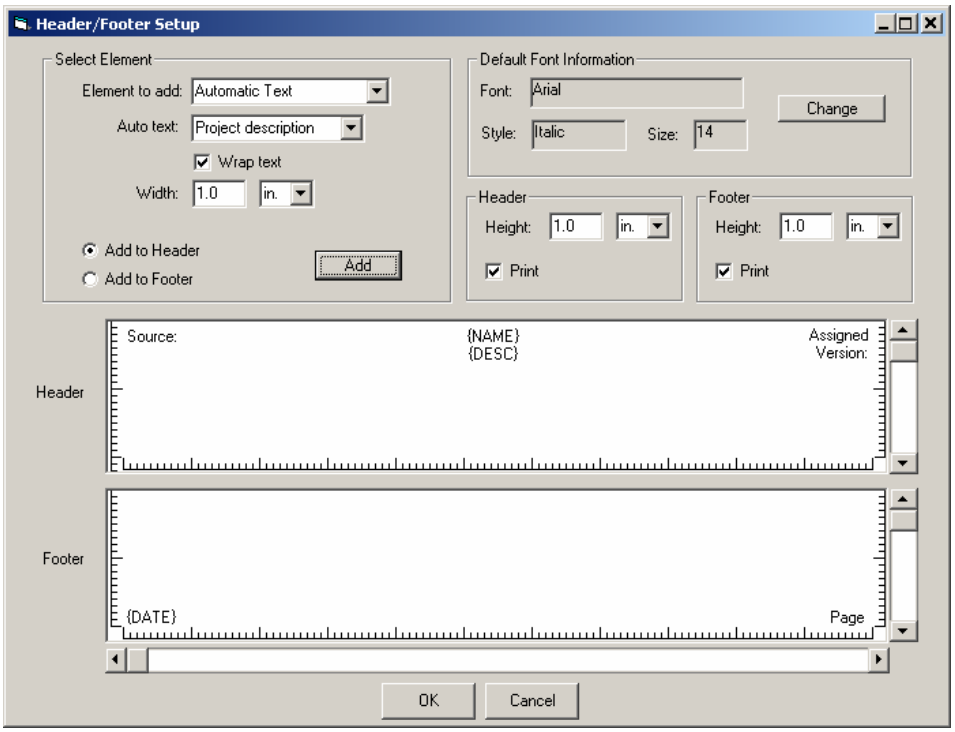
The set of icons that are used for display are limited to a few different graphic symbols. This set of icons remains the same regardless if any objects in the trees have been assigned a UDS. This means that no special icons are used to distinguish those objects in the trees that have a UDS assigned to them, nor can the icons be changed. The assignment of a UDS only affects the display and printing of a chart.



Sample icons displayed in tree view

Specifying chart header and footer layouts

Capabilities are available to format the layout of the header and footer that are included with the chart printout. These include flexibility in defining the font used for printing (selectable for each text element), the location of the element on the page, and additional elements that can be included.



Header/footer layout dialog

Select element frame

The Select Element frame determines what type of element will be added to the header or footer. The available elements in the "Elements to add" list are:

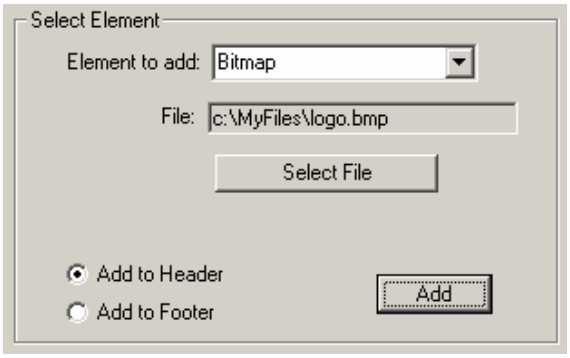
- Bitmap
- Automatic text
- User text

- Vertical line
- Horizontal line
- Rectangle
- Chart Border

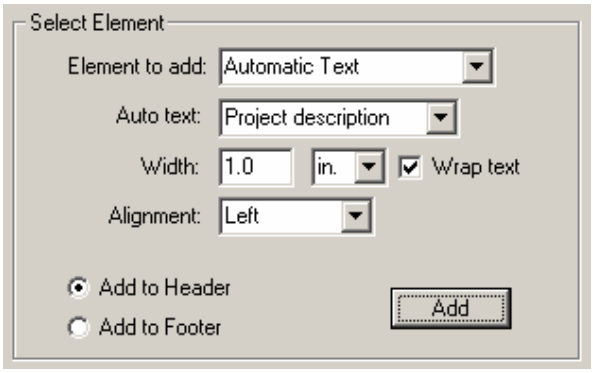
Depending on the type of element selected, the contents of the frame will change to allow the entry of necessary information for that element. Examples of each element's definition are shown in the following Figures.

Select where the element is to be added, either in the header or in the footer, and then when satisfied with the setup, select "Add". The element is then added to the specified section of the report (where it can then be located to the precise location).

The exception to this process is the "Chart Border" element. This element is not added, but simply specified whether it is to be printed or not. If it is to be printed, then specify if the border is to be placed just around the chart area, or placed around the header and chart, the footer and chart, or the entire page.



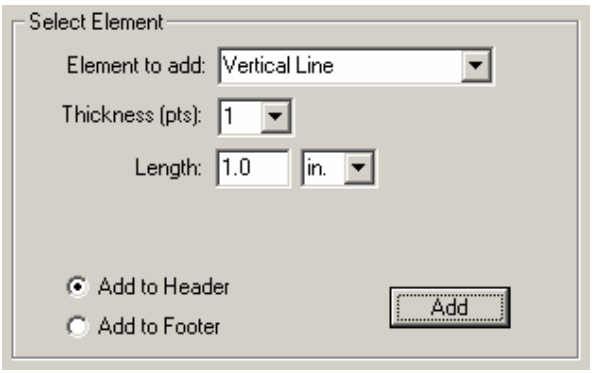
Definition of a bitmap



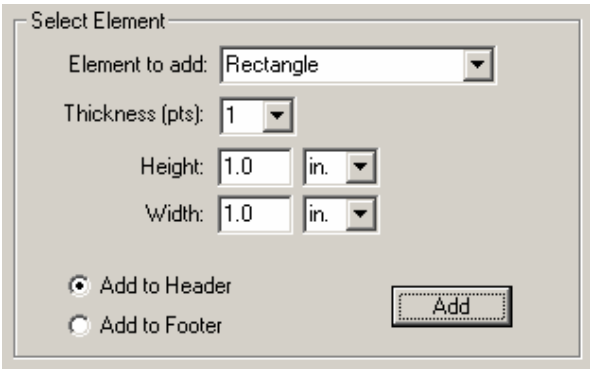
Definition of automatic text



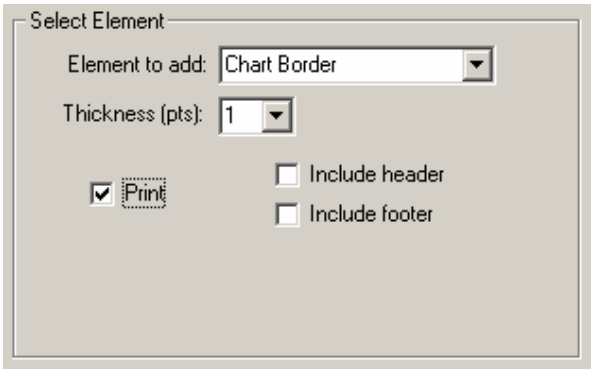
Definition of user text



Definition of a line



Definition of a rectangle



Definition of the chart border

Default font information frame

When an automatic text or user text element is added, the Default Font Information frame will be displayed with these settings.

To change the current default, select "Change", and the standard font selection dialog will be displayed. It is possible to change the font for a text element once it has been added by displaying the properties dialog.

Header and footer frames

The header and footer definitions determine if the header or footer will be printed, and what the height of each one is, in either inches or centimeters. Note that you can have elements defined in the header or footer, and if "Print" is not selected, that information will not be printed.

Defined elements display

This area is where defined elements are displayed, and where they can be positioned by dragging them to the desired location.

To delete an element from the report, select it and press the **Delete** key on the keyboard.

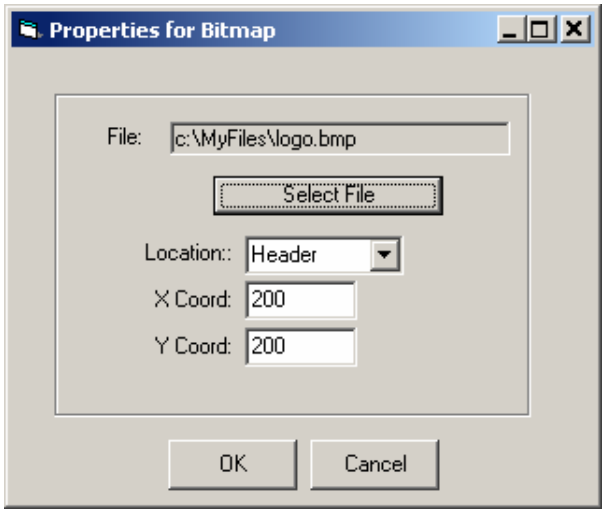
To move an element to another location within its specified header or footer, perform a drag and drop operation.

To change an element's properties such as font, size, etc., right-click on it and the properties dialog for that element will be displayed. The properties dialogs allow the user to change the properties that were assigned when the element was added, and they also

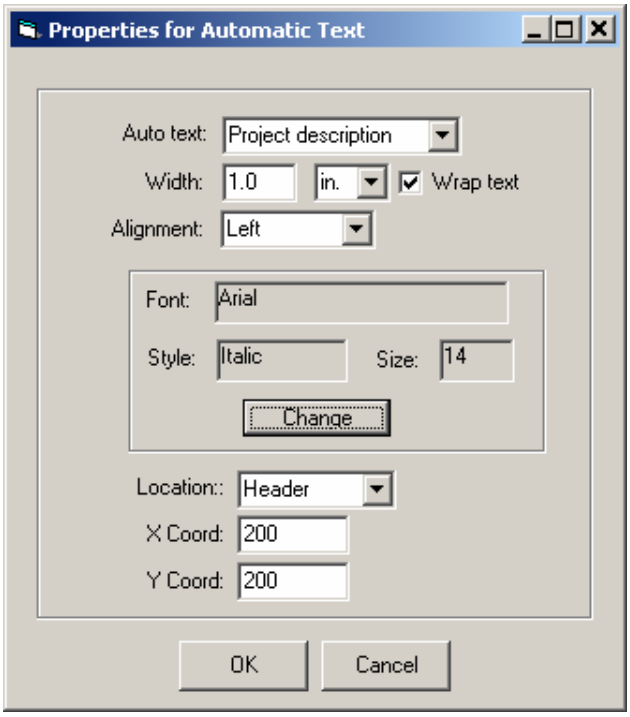
allow precise positioning of the element. Examples of each of the properties dialogs are shown the following Figures.

For automatic text and user text, option **wrap text** can be checked. If this option is checked, then the user specifies the width of the text box in which the text will be displayed. If the specified text is longer than the space defined, the text will be split into another line immediately below the first line. This splitting will continue until all of the text is printed. The location where the text is split is where a space exists. If no space exists in the text within the width of the text box, the text is split at the location that fills the width. If **wrap text** is not specified, the text will be printed on one line, and if the text is long, overprinting may occur with other elements. Care must be taken to ensure that text is not allowed to overprint to the side of, or below the current element.

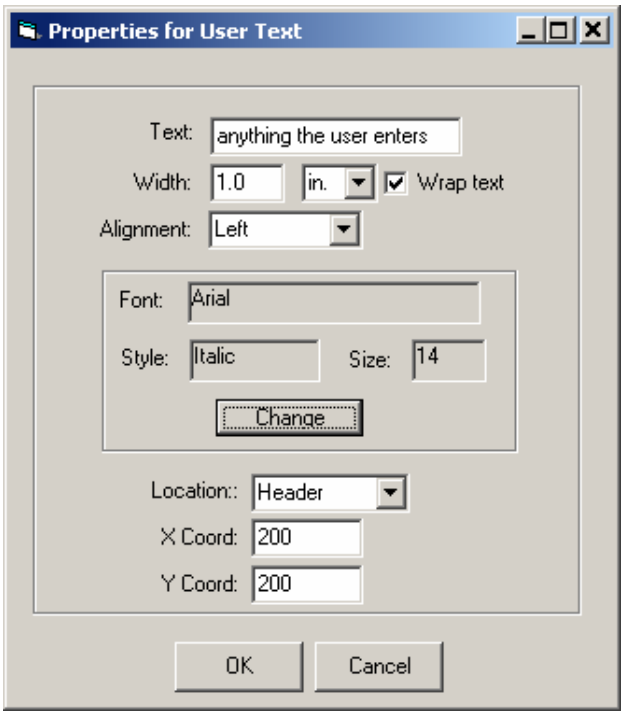
Text alignment can be specified for left, center, or right. For fields that do not have **wrap text** specified the length of the field expands to accommodate the text. When **left** alignment is selected, the field will expand to the right. When **right** is selected, the field will grow to the left. When **center** is specified, the field will grow equally to the left and the right. If **wrap text** is specified, then the alignment of the text takes place within the width of the field.



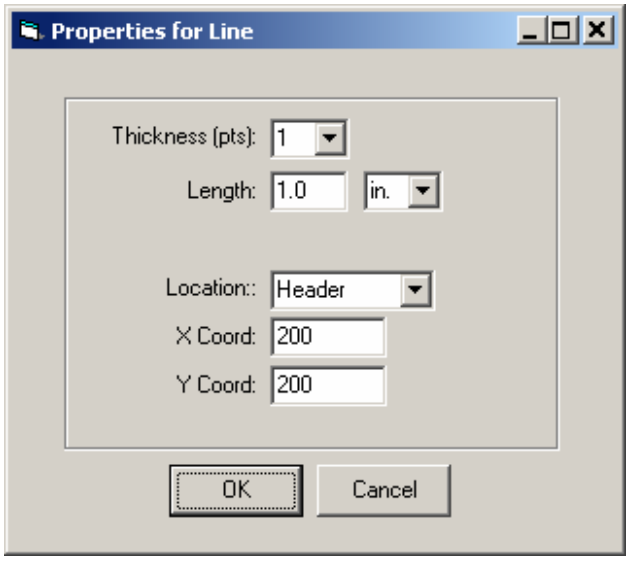
Properties for Bitmap dialog



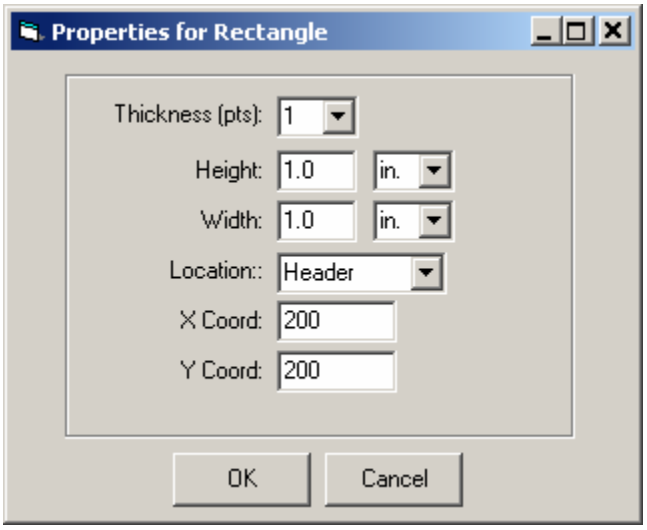
Properties for Automatic Text



Properties for User Text



Properties for Line



Properties for Rectangle

Printing a chart

A single chart or multiple charts can be printed by highlighting it in the tree or having it open, and then selecting **File** > **Print** > **Charts** from the menu, or pressing Ctrl +P.

File > **Print** > **All Project Tree Charts** and **File** > **Print** > **All Monitoring Tree Charts** are also supported.

Charts print at the same scale that is specified for its display (as selected by the drop down scale factor on the Control Builder tool bar). The scale that the chart is printed at does not change the scale of the header or footer, or any information in the header and footer. Also, if the **All Project Tree Charts** or **All Monitoring Tree Charts** print option is selected, all of the charts are printed at 100%. The choices for the "zoom" factor are 500%, 300%, 200%, 150%, 120%, 100%, 80%, 50%, and 20%.

Appendix F - Control Builder Checkpoint Reference

Using Checkpoint to Save and Restore Data

The Checkpoint function lets you save and restore operation and configuration data associated with a controller. In this document, we also refer to a controller as a top-level-hardware-parent node (TLHPN) or just *node*. The primary purpose of Checkpoint's save and restore functions are to provide the operator with the capability to bring a failed controller back to an operational condition as soon as possible. There are other ancillary scheduling, archiving and rebuilding functions provided by Checkpoint.

The Checkpoint function does **not** backup and restore you Experion system and data. For information about the Experion Backup and Restore software, please refer to the *Backup and Restore Guide* in Knowledge Builder.

This section includes topics to explain what the Checkpoint function is and the tasks associated with using it. The Topics listed in the following table are links to the given information location in this section. Just click the topic to activate the link, when viewing the contents online.

Topic/Link
Error! Reference source not found.
Checkpoint disk space maintenance functional considerations
Reviewing Checkpoint Functional Aspects
Checkpoint file characteristics
Control Builder Checkpoint interface summary
Station Detail Display Checkpoint interface summary
System Status Display Checkpoint interface summary
System Event Checkpoint interface summary
Identifying Functions on the Checkpoint Scheduler Dialog
Error! Reference source not found.
Error! Reference source not found.
Checkpoint scheduler graphical reference
Identifying Functions on the Archive Checkpoint Files Dialog

Topic/Link
Archiving Checkpoint Files
Archive checkpoint graphical reference
Checkpoint file attributes
Error! Reference source not found.
Pre-selection of nodes to be saved
Save arbitration on a given Node
Execution states for checkpoint save
Initiating Manual Checkpoint Save
Save checkpoint manually graphical reference
Identifying Functions on the Restore from Checkpoint Dialog
Pre-selection of nodes to restore
Restore scope selection and actual restore action
Restoration file display reference
Order of restore checks
Execution states for checkpoint restore
Restoring From Checkpoint
Restore from checkpoint graphical reference
Rebuilding Checkpoint from Monitoring Tab
Using Detail Displays for checkpoint operations
Checkpoint operations through Detail Displays graphical reference
Troubleshooting Checkpoint Function
Viewing error logs
Identifying checkpoint errors logged but not journaled
Checking messages journaled as events
Checking warnings and errors Logged but not journaled as events

Topic/Link
Checkpoint operation with On-Process Migration

Getting Started

This section provides some overall background information before you begin using the Checkpoint function.

Checking status of Engineering Tools application

Prerequisites:

- You have logged on with Administrative Privileges.
- You have installed Experion software version R300.1 or greater.
- The Windows Task Scheduler service is running. (This service starts automatically and should never require any user interaction during normal operation.)

Considerations:

- You are familiar with navigating within a Windows environment.
- You can use common shortcut key (Alt+Key) actions to access all the enabled fields/controls on the Checkpoint user interface.
- The Checkpoint service is only installed on the primary and secondary Experion Servers.
- The System Repository (SR) service starts the Checkpoint service on the primary server and stops the Checkpoint service (and keeps it stopped) on the secondary server.
- Do **not** change the running/stopped state of the checkpoint service or any service properties related to the checkpoint service.
- You **cannot** initiate a manual or automatic checkpoint save function when the checkpoint service is not running. If you attempt any operation from the Checkpoint user interface (UI) that interacts with the Checkpoint Service when the service is not running, a subsequent timeout connection messages is generated.
- When the Checkpoint service is stopped, the scheduled **Automatic Save** tasks will **not** run, so checkpoint files will **not** be created. The scheduled tasks resume operation, when the Checkpoint service is re-started by the System Repository (SR).

- If the actual primary server's File Replication service is stopped, checkpoint file replication does **not** occur to the actual secondary server or to any of the console stations.
- When Control Builder or Detail Displays are closed, the Checkpoint service on the actual primary server keeps running in the background to checkpoint controllers listed in the scheduled (automatic) tasks.

Use the following procedure to confirm that all the Engineering Tools related services are running. This is optional since all applicable Experion services start automatically when the computer is started.

To check Engineering Tools status

Step	Action
1	Click the **Start** button, and then point to **Programs>Honeywell Experion>Engineering Tools** and click **Engr Tools Services Control Panel**.
2	In the **Stop_Start_Services_Dialog**, click the **START ALL including SQL** selection to select it.
3	Click the **OK** button and monitor the progress of the operation in the open **Start_All_Services_Application** dialog. The services are started when the dialog closes.
4	This completes the procedure.

Configuring preferences for Checkpoint function

Prerequisites:

- You have logged on with Administrative Privileges.
- You have installed Experion software version R300.1 or greater.
- You have launched Control Builder through the Configuration Studio.
- All applicable Experion services are running.

Considerations:

- You are familiar with navigating within a Windows environment.
- You are familiar with the Control Builder application.

- A user with a security level of Manager (Mngr) has access to all Checkpoint functions by default.
- Configured preferences are stored in the Engineering Repository Database (ERDB) and the System Repository (SR).
- The gigabyte (GB) is the unit of measurement used to specify the amount of free disk space that checkpoint is not allowed to consume for checkpoint files.
- You can configure the following preferences.

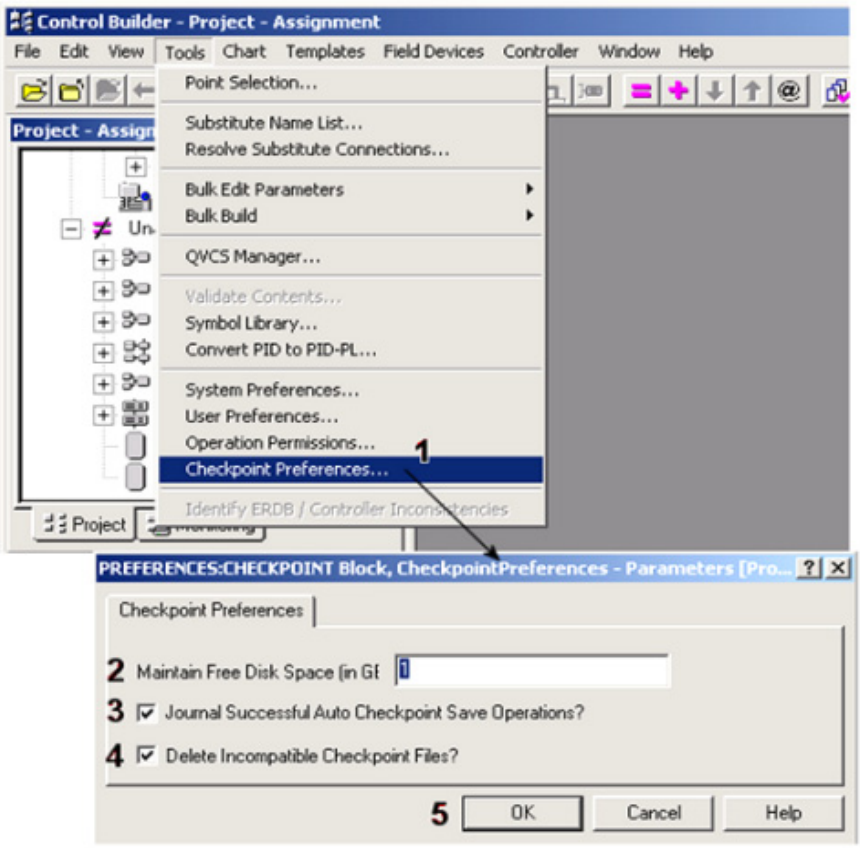
For This Preference . . .	You Can . . .
Maintain Free Disk Space (GB) of:	Define the free disk space size in Gigabytes (GB)) that has to be maintained at all times. If the disk space drops below the threshold size set in this preference, an event is logged. The default setting is 1 GB.
Journal Successful Auto Checkpoint Save Operations?	Choose to enable or disable the generation/journaling of successful checkpoint save events for each node that is saved by an automatic checkpoint task. When disabled, periodic successful save events are suppressed to avoid clogging the journals with many save messages. The default selection is unchecked, which means that the successful save messages will not be journaled. • This function does not suppress the periodic start and stop events associated with an autocheckpoint task, these events are always journaled even when this function is disabled. • This function does **not** suppress successful save messages produced from either **Save Checkpoint Manually** selections of the node(s), or from successfully saved checkpoint files produced from execution of **Save Checkpoint Manually** on manual tasks. The successful save messages are always journaled for any manual save operation. • This checkpoint preference also does **not** apply to the suppression of any failed save message. Failed saved events, for both **automatic** and **manual** checkpoint saves, are always sent to the system event journal.
Delete Incompatible Checkpoint Files?	Choose to delete all the incompatible checkpoint files of a node when it is being loaded or not. The default is checked, which means that incompatible checkpoint

For This Preference . . .	You Can . . .
	files will be deleted.

To configure checkpoint preferences

Step	Action
1	On the **Tools** menu, click **Checkpoint Preferences**.
2	On the **PREFERENCES:CHECKPOINT Block** dialog, type desired size in gigabytes for the free disk space to be maintained after every Checkpoint save operation in the **Maintain Free Disk Space (in GB) Of:** field. The default setting is 1 (GB). Press the **Tab** key.
3	Select the **Journal Successful Auto Checkpoint Save Operations?** check box, if you want a journal message triggered for every successful save operation. Otherwise, leave the checkbox blank to suppress journaling of successful Checkpoint save operations. A message is sent for every failed save operation. The default setting is a blank check box. Press the **Tab** key.
4	Select the **Delete Incompatible Checkpoint Files?** check box, if you want all the incompatible checkpoint files of a hardware node deleted when the node is being loaded. Otherwise, be sure the check box is blank. The default setting is checked, which means incompatible checkpoint files will be deleted when the node is being loaded.
5	Click the **OK** button to save the changes and close the **Preferences** dialog box.
6	This completes the procedure

Checkpoint preferences procedural illustration reference



Configuring Checkpoint Preferences

Configuring operation permissions for Checkpoint functions

Prerequisites:

- You have logged on with Administrative Privileges.
- You have installed Experion software version R300.1 or greater.
- You have launched Control Builder through the Configuration Studio.
- All applicable Engineering Tools services are running.

Considerations:

- You are familiar with navigating within a Windows environment.
- You are familiar with the Control Builder application.
- A user with a security level of Manager (Mngr) has access to all Checkpoint functions by default.
- Configured Operation Permission preferences are stored in the Engineering Repository Database (ERDB) and the System Repository (SR).
- A user with the permission level specified for the **Set Preference/Permission** operation in the **Preferences** dialog can configure which of the following Checkpoint functions a user with a lower security level will be permitted to access.

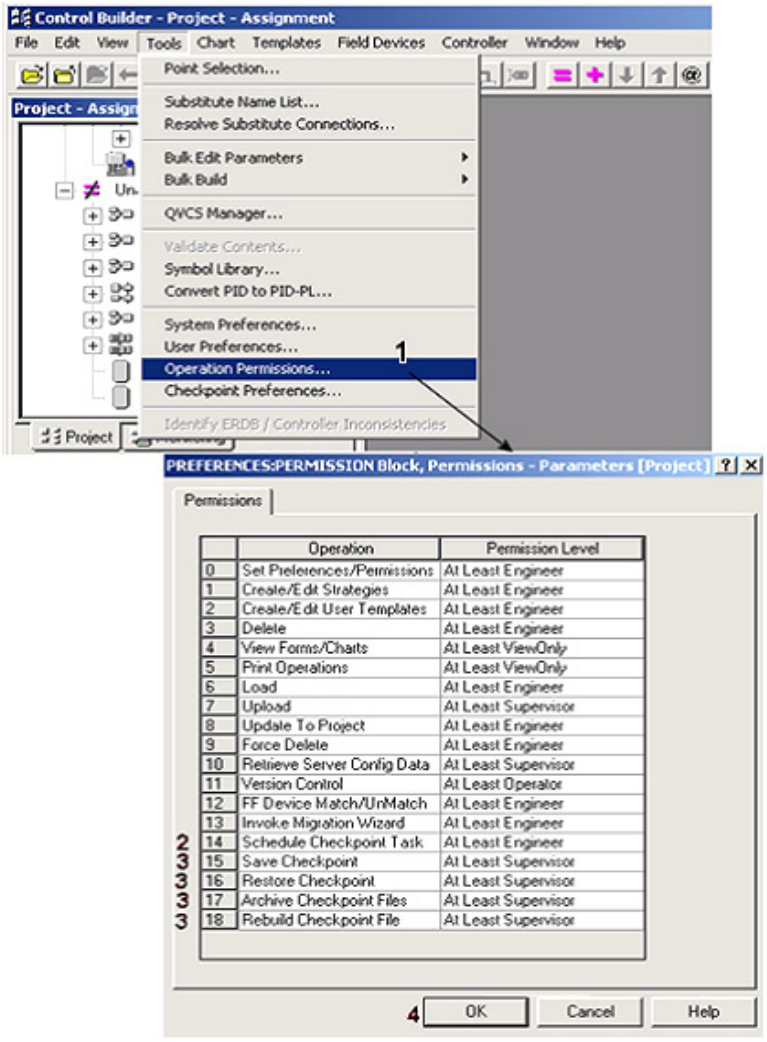
If Checkpoint Function is . . .	Then, User Can . . .
Schedule Checkpoint Task	Create both periodic and manual tasks that perform checkpoint saves of selected controller nodes. For the periodic tasks, schedule the checkpoint save at specified time intervals.
Save Checkpoint	Manually initiate a Checkpoint save.
Restore Checkpoint	Manually initiate a Checkpoint restore to return a node to a previously stored configuration/operational condition sufficient for a warm restart.
Archive Checkpoint Files	Archive selected checkpoint files to another location and/or medium, with the option of making the archived file(s) read only.

If Checkpoint Function is . . .	Then, User Can . . .
Rebuild Checkpoint File	Rebuild Checkpoint from Monitor tab to rebuild the entire checkpoint base associated with given node.

To configure operation permissions

Step	Action
1	On the **Tools** menu, click **Operation Permissions**.
2	On the **PREFERENCES:PERMISSION Block** dialog, find the **Schedule Checkpoint Task** in the **Operation** column, click the corresponding row in the **Permission Level** column, and select the desired permission level from the list. The default setting is **At Least Engineer**.
3	Repeat Step 2 to select the desired permission level for these other Checkpoint operations: • **Save Checkpoint** • **Restore Checkpoint** • **Archive Checkpoint Files** • **Rebuild Checkpoint File** The default setting for these operations is **At Least Supervisor**.
4	Click the **OK** button to save the changes and close the dialog
5	This completes the procedure.

Checkpoint preferences procedural illustration reference



Configuring Operation Permissions

Checkpoint disk space maintenance functional considerations

The Checkpoint function attempts to maintain the previously configured free disk space size after every save operation. The following table summarizes some functional considerations for how this space is conserved and consumed.



TIP

It is a good idea to archive checkpoint files and back up a compatible ERDB at Project Engineering Milestones. This ensures that the checkpoint structure information remains consistent with the monitoring ERDB and backup of the ERDB requires that you manually backup the Checkpoint Base share that is on the primary server.

If Checkpoint Operation Is For. . .	Then, Consider This. . .
Checkpoint file being created for a **Manual** save operation, including those done by Manual tasks.	• The checkpoint file is written to the disk regardless of the free space restriction. If this saving results in the violation of the free space requirement, an event is generated. • If the checkpoint file cannot be created because of the unavailability of disk space, the message "`Not able to create checkpoint file, insufficient disk space.`" is displayed and an event is generated.
Checkpoint file being created for an **Autocheckpoint** save	• If the file being saved is the first auto-checkpoint file for a node, which means there are no previous versions of checkpoint files for this given autocheckpoint task, – The file is written to the disk regardless of the free space restriction. – If this save results in violation of the disk free space requirement, an event is generated. – If the available disk space is inadequate, an event is generated and the checkpoint file is not created. Also, the message "**`Not able to create checkpoint file, insufficient disk space.`**" is displayed and an event is generated. • If there are previous versions of the checkpoint file for

If Checkpoint Operation Is For. . .	Then, Consider This. . .
	a node in this checkpoint task and the free space limit has been reached, – The oldest version is deleted and the new version is written to the disk. – Before saving the new file, it is ensured that the space regained by deleting the old file meets the disk space required by the new file – If this space is inadequate, then the older versions are deleted until there is enough disk space to save the new file. This reduces the number of versions of the checkpoint file for this node. – At least 1 version of the checkpoint file is maintained
Deleting incompatible files when associated checkpoint preference is enabled	• Any **Project** load and delete operation will cause incompatibility, and disqualify all prior existing checkpoint files for use in any checkpoint restore operation. (This does **not** apply to the node's latest.cp, as that always remains compatible.) A monitor load does not cause incompatibility. • If the checkpoint preference '**Delete Incompatible Checkpoint Files**' is checked/enabled, all incompatible checkpoint files located within the default checkpoint folders are deleted soon after a load operation. Checkpoint files, which are located outside the default folders, are not deleted. The default folders being referred to are those subdirectories under the node's directory which is under the Checkpoint share.

Reviewing Checkpoint Functional Aspects

This section reviews some of the underlying functional aspects of the Checkpoint operation.

Functional Description

The Checkpoint function captures and stores configuration or static data and operational or dynamic data for the following hardware nodes and their contents in checkpoint data files.

- C200 Controller
- C300 Controller
- Application Control Environment (ACE) Node
- Simulation Control Environment (SCE) Node - SIM-C200, SIM-ACE, and SIM-IOLIM.
- Legacy I/O Module (LIOM)
- OPC Gateway
- Inter Cluster Gateway (ICG)

Checkpoint file characteristics

A checkpoint file contains a proprietary binary image of the following data captured from a given node, as applicable. It also includes an attribute file for general definitions and compatibility comparisons. The checkpoint files are stored on the computer's local drive. The checkpoint files are created on the Primary Server in the default checkpoint share directory. This share is then replicated to the secondary server and each console station in that server cluster.

Node Data	Description
Configuration Checkpoint Data (CCD)	CCD consists of Load Only information and load information that can change at run time. Checkpoint acquires this parameter information during the project/monitor load operation of a node. This information also gets acquired by Checkpoint when a user initiates a checkpoint rebuild operation.
Operational Checkpoint Data (OCD)	OCD information consists of those loaded parameters that can change during runtime. As such, a loaded parameter may have both CCD and OCD attributes. However, OCD information also consists of non-loaded parameters, such as execution states, that change during runtime. Checkpoint reads this OCD information from the hardware node and all its hardware/software child blocks during a checkpoint save. During a **Checkpoint Restore** operation, parameters are restored according to the load order. Even non-loaded parameters (such as execution states) have an implicit load order during the checkpoint restore. The full complement of the node's restored CCD/OCD data permits a warm restart of the hardware node, if supported.

Node Data	Description
Control Component Language (CCL) Libraries	When blocks using CCLs are loaded to the node, then the CCL library names, library code, and firmware version information are stored in the checkpoint file. This CCL information is stored in the checkpoint file so that checkpoint restore can be done without requiring access to an operational ERDB. During a **Checkpoint Restore** operation, the firmware version is retrieved from the controller and compared with the firmware version in the checkpoint file. If there is a firmware mismatch, the restore will not be successful.
Custom Algorithm Block (CAB) Algorithms	Similar to CCL requirements, CAB related information is stored in the checkpoint file so that checkpoint restore can be done without requiring access to an operational ERDB. Checkpoint captures CAB configuration parameters including the algorithm during the load of the block instance. If there are multiple instances of a CAB type, only one copy of the algorithm is stored and the same copy is referenced by all the instances. The algorithm is retained as long as there is at least one instance of a CAB type referencing it. The algorithm is deleted once the only block referencing it is deleted. During a Checkpoint Restore operation, the algorithm is extracted from the checkpoint file and the parameters are returned in the same order and format that they were stored during a load operation.

Checkpoint files and files storage overview

The following table provides an overview of the shared Checkpoint and CheckpointBase folders/directories associated with checkpoint files and their storage.



ATTENTION

You cannot choose the storage location for manual checkpoint saves. During a **Save Checkpoint Manually** operation, checkpoint saves to certain subdirectories under the following default share folder/directory:

\\primary_server\%systemdrive%\Documents and Settings\All Users\Application Data\Honeywell\Experion PKS\Checkpoint

Only Checkpoint Files in this default folder are replicated to the secondary server or to the console stations.

Function	Checkpoint Folder	CheckpointBase Folder
Purpose	Checkpoint Files are created both automatically on a scheduled basis and manually using the checkpoint scheduler. These files are stored by the Checkpoint Service under the default shared Checkpoint folder/directory in the primary Experion server. Be **aware** that you must never modify or delete the node's latest.cp from this Checkpoint share, and should never modify any subdirectory names in this share, since it may introduce problems in checkpoint save operations.	To create/save checkpoint files for a given node, checkpoint must acquire the configuration information associated with that hardware node. The Checkpoint Service only reads the run-time changeable information from the hardware node. The checkpoint functionality need only acquire the non-changeable configuration information at a node's load time. The configuration information held by checkpoint also gets modified when a tagged object within that node gets deleted/reloaded/uploaded. This configuration information, along with other file management information, is captured and maintained by Checkpoint in the node's subdirectory of the shared CheckpointBase folder. The Checkpoint Base share only exists in the primary/secondary server file system. This Checkpoint Base share is replicated between the primary and secondary server during project/monitor loads, deletes, and checkpoint rebuilds. Checkpoint saves also update this base information so it remains in sync with what is found in the node's latest.cp. Be **aware** that you must never modify/delete any files in this Checkpoint Base share
Hardware Node Files	A folder for each hardware node is created under this shared Checkpoint folder in the primary Experion server. Further sub-folders are under each node folder. The checkpoint files are stored in these sub-folders. See the following section for more details about the directory structure related to this share. This entire shared	Like the Checkpoint share, a folder for each hardware node is created under the shared CheckpointBase folder. However, it has no sub-folders under each node folder. The checkpoint structure and management files are stored directly in the node folder. As noted in the preceding row, this entire CheckpointBase share is replicated to

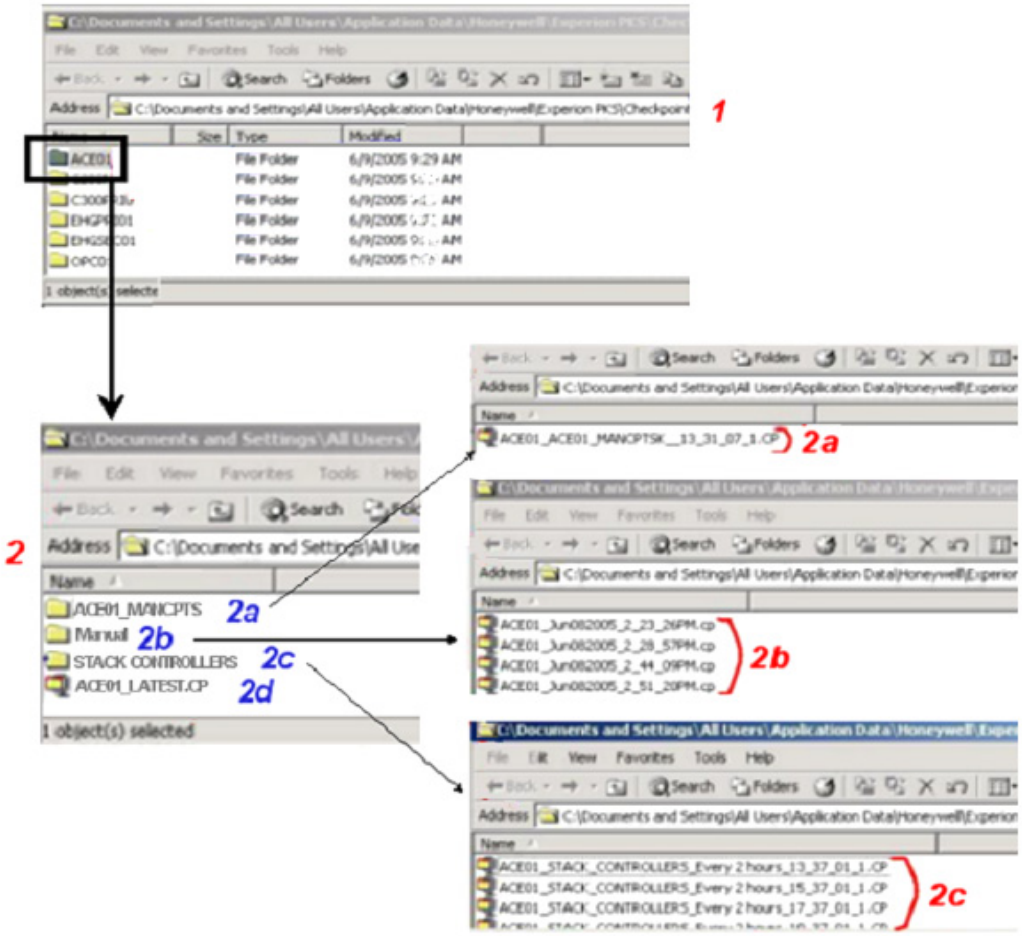
Function	Checkpoint Folder	CheckpointBase Folder
	Checkpoint folder is then replicated to the secondary server as well as to all Console Stations.	only the secondary server.
Storage	As stated in the preceding Attention note, checkpoint saves can only be made to the local drive's default directory of the shared Checkpoint folder on the primary server, with automatic replication to that same named default directory for Checkpoint shares on the secondary server, and each console station's local drives. However, the Checkpoint Service gives users the capability to archive checkpoint files from these Checkpoint shares to not just the destinations on these same local drives, but also to other machine's, other local/network drives, and to removable media, such as a CD or DVD. Checkpoint also provides the ability to restore a node directly from a checkpoint file from any of these destinations where the archive was directed.	

Checkpoint functional rules and guidelines

- All checkpoint files will be created in the following Checkpoint share directory. See the following figure for sample default directories and files.
  - *C:\Documents and Settings\All Users\Application Data\Honeywell\Experion PKS\Checkpoint*
- For a given node, the checkpoint files will be placed in a node sub-directory located in the Checkpoint share. This node sub-directory will be named with the same character string as in the Experion tag name of that node. For documentation purposes, below that string will be referred to as **ControllerName**
- All checkpoint files will have this extension: cp.

- Each node will have a latest checkpoint file named as **ControllerName_Latest.cp** . This checkpoint file will be located in the node's subdirectory in the Checkpoint share.
- All checkpoint files created from a manual save operation will have default names based on the following convention. The time and date is obtained from the primary scan server's local time base.
  - **ControllerName_Date_Time.cp**
  - For example, ACE01_JAN042004_5_34AM.cp
- All checkpoint files created from a scheduled autocheckpoint task will be named based on the following convention:
  - **ControllerName_Task_Interval_Version.cp**
  - For example, ACE01_TaskName_daily_01.cp
- All Checkpoint Tasks scheduled to run **automatically** will be named based on the following convention:
  - **ControllerName_AutoCpTsk**
  - For example, ACE01_AutoCpTsk
- All Checkpoint Tasks scheduled to run **manually** will be named based on the following convention:
  - **Controller Name_ManCpTsk**
  - For example, ACE01_ManCpTsk
- The manually saved checkpoint files will be placed into subdirectories located under the node's overall subdirectory in the Checkpoint share.
  - If the save was initiated by a manual checkpoint task, the saved checkpoint file will be placed into a subdirectory named the same as the manual task's name.
  - If the saved was initiated on that node, other than by a manual task, then the saved checkpoint file will be placed into the common "Manual" subdirectory.
- All checkpoint files saved by a given autocheckpoint schedule will be placed into a subdirectory under the node's overall subdirectory in the Checkpoint Share. That subdirectory (having the autocheckpoint checkpoint files) will have the same name as that autocheckpoint task's name.

- Microsoft's cabinet file format is the underlying technology used for producing the checkpoint files.
- Microsoft's makecab.exe utility compresses the information during the checkpoint save operation. During the checkpoint restore operation, the extract.exe is used to decompress the information
- All date and time fields display local date and time. The time and date is obtained from the primary server's local time base.



Sample Default Directories and Files

Callout	Description
1	Top level of share has subdirectory for each node.
2	Each node subdirectory has:
2a	Subdirectory of each manual checkpoint task. Contents of given subdirectory are the node's checkpoint files saved from launches of the manual task on the Checkpoint Save display's **By Task** tab.
2b	Common Manual subdirectory containing checkpoint files of manual saves launched for this node from the **By Node** tab on the Checkpoint Save display.
2c	Subdirectory for each automatic checkpoint task. Contents of each subdirectory are the checkpoint files saved as result of periodic execution of that task.
2d	The node's latest.cp checkpoint file.

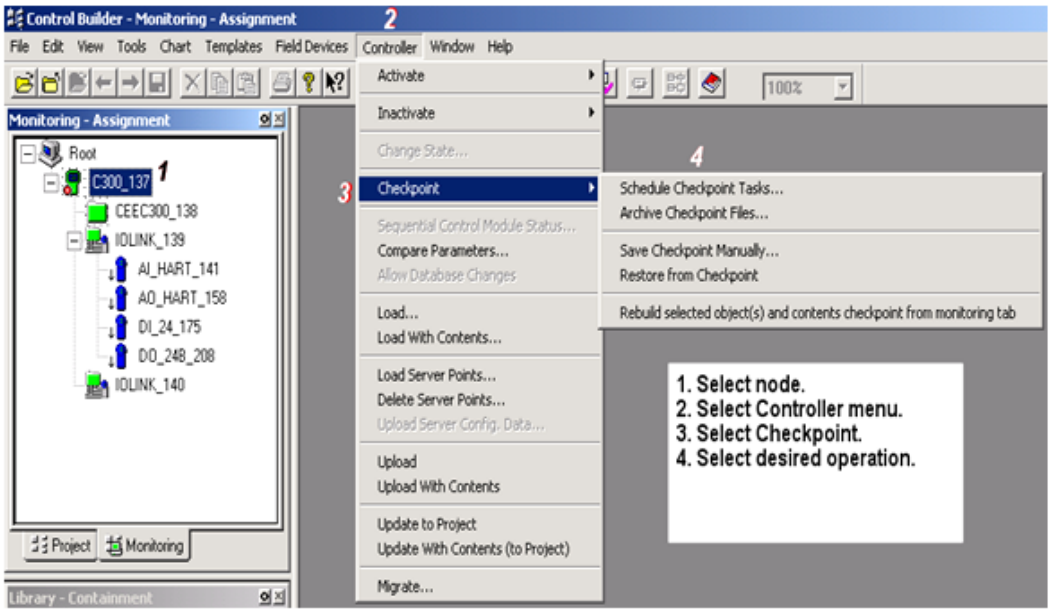
Control Builder Checkpoint interface summary

The following table summarizes the checkpoint operations that can be launched from Control Builder.

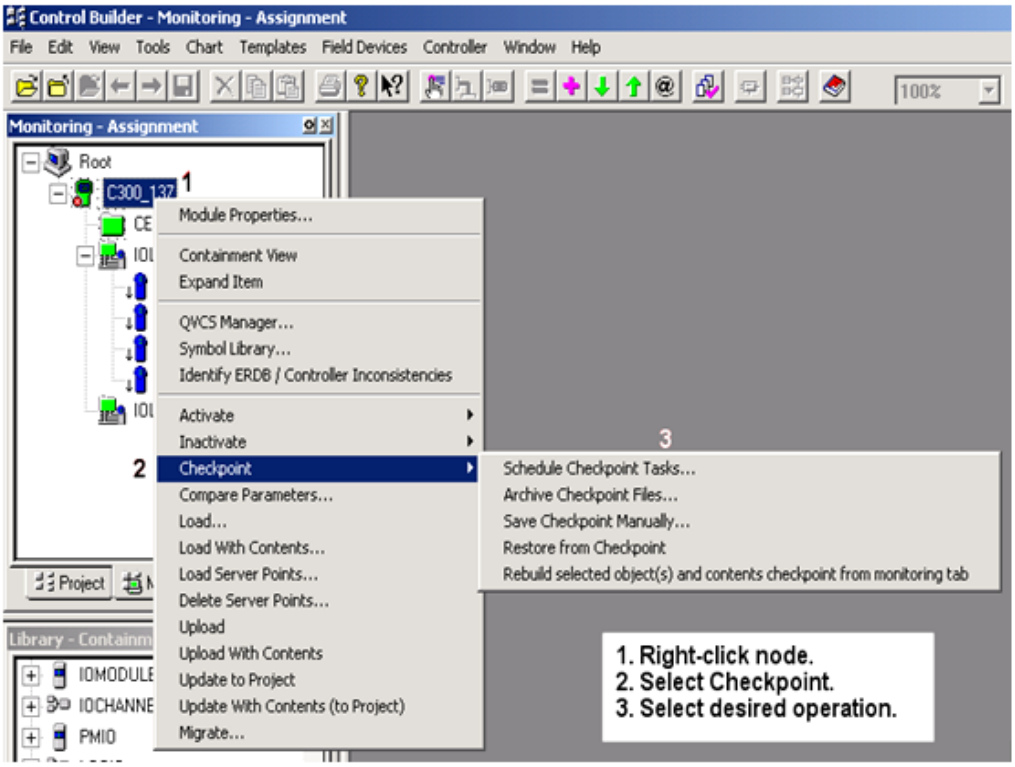
Initiate This Operation . . .	If You Want To . . .
On **Controller** menu or right-click node, click **Checkpoint**>**Schedule Checkpoint Tasks**	Create tasks that periodically checkpoint all the nodes listed in the task. • The complete functionality of the **Checkpoint Scheduler** dialog can only be launched from Control Builder.
On **Controller** menu or right-click node, click **Checkpoint**>**Archive Checkpoint Files**	Archive desired set of checkpoint files to a different location.
On **Controller** menu or right-click node, click **Checkpoint**>**Save Checkpoint Manually**	Checkpoint a selected node and store the current configuration and operational data. In addition, you can save a group of nodes based on previously configured manual tasks.
On **Controller** menu or right-click node, click **Checkpoint**>**Restore**	Restore a failed hardware node or return to a previous configuration/operational condition by choosing a checkpoint file from among multiple checkpoint files. For certain nodes, you can

Initiate This Operation . . .	If You Want To . . .
From Checkpoint	selectively restore lower level hardware modules.
On **Controller** menu or right-click node, click **Checkpoint> Rebuild Selected Object(s) and Contents Checkpoint from Monitoring Tab**	Re-construct both the checkpoint base information (CCD info) of that node, and regenerate the node's Latest.cp checkpoint file with just the configuration information. The information for doing this is obtained from that node's loaded information in the Monitor side of the ERDB.

Control Builder Interface graphical reference



Initiating Checkpoint operations through CB Controller menu



Initiating Checkpoint operations through CB right-click menu

Station Detail Display Checkpoint interface summary

The following table summarizes the Checkpoint operations that you can access through Detail Displays in Station. These are subsets of the Checkpoint operations that are available through Control Builder.

If You Want To . . .	Then, View Checkpoint Tab On . . .	And Use Interface To . . .
View Checkpoint Task Status	Node's Top Level Display	Examine the current status of all the checkpoint tasks scheduled for a node. This is a view only display, since you are not allowed to modify the tasks.
Save Checkpoint Manually	Node's Top Level Display	Checkpoint save a selected node. You can only save individual nodes from detail displays, and cannot save based on manual tasks. Detail Display does not give you the option to perform a Manual Task save.
Restore From Checkpoint	Node's Top Level Display or Lower Level Hardware Item's Display	Launch same checkpoint restore operations from a node's detail display and lower level module detail displays (as appropriate) as you would from Control Builder.

System Status Display Checkpoint interface summary

You cannot directly interface with Checkpoint operations through System Status Display. You can launch a node's detail display from the System Status Display and then interface with Checkpoint operations through the Checkpoint tab on the detail display.

System Event Checkpoint interface summary

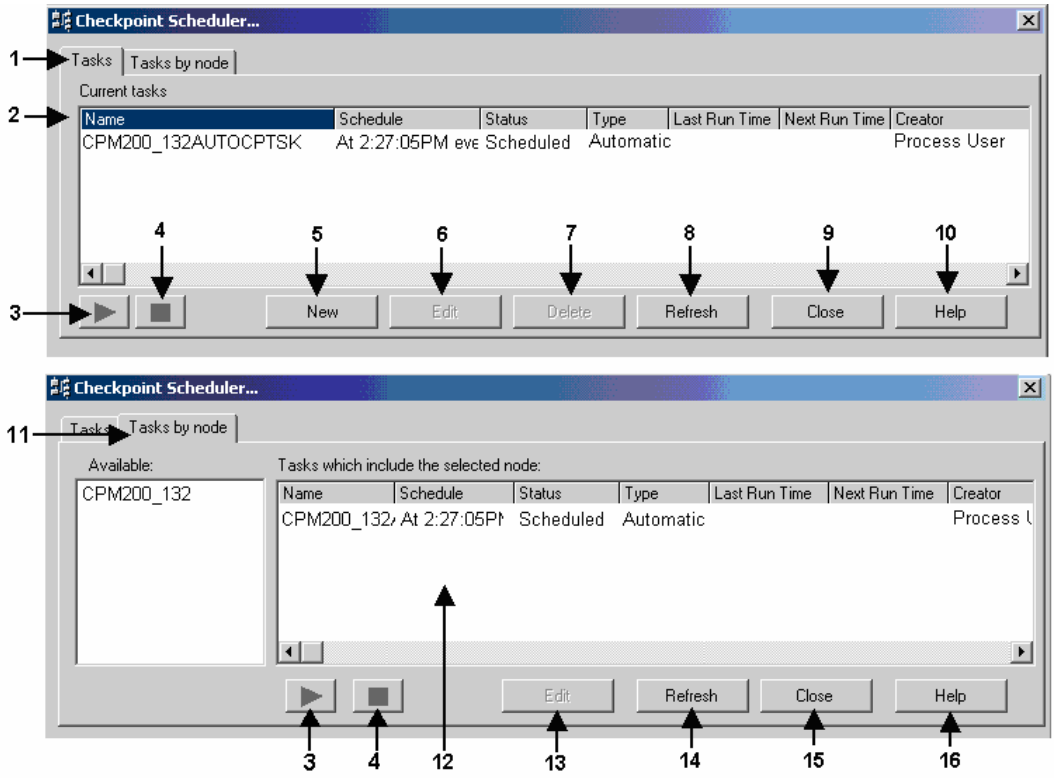
You can call up the server's system Event Journal through Station to view Checkpoint related events being journaled in the server's Event system.

Identifying Functions on the Checkpoint Scheduler Dialog

The following illustration and callout table identify the functions that are available on the **Checkpoint Scheduler** dialog.

TIP

You can change the viewing order of data from ascending to descending or vice versa by clicking the cursor in a column heading in the table grid on the dialog.



Checkpoint Scheduler Task and Task By Node Tabs

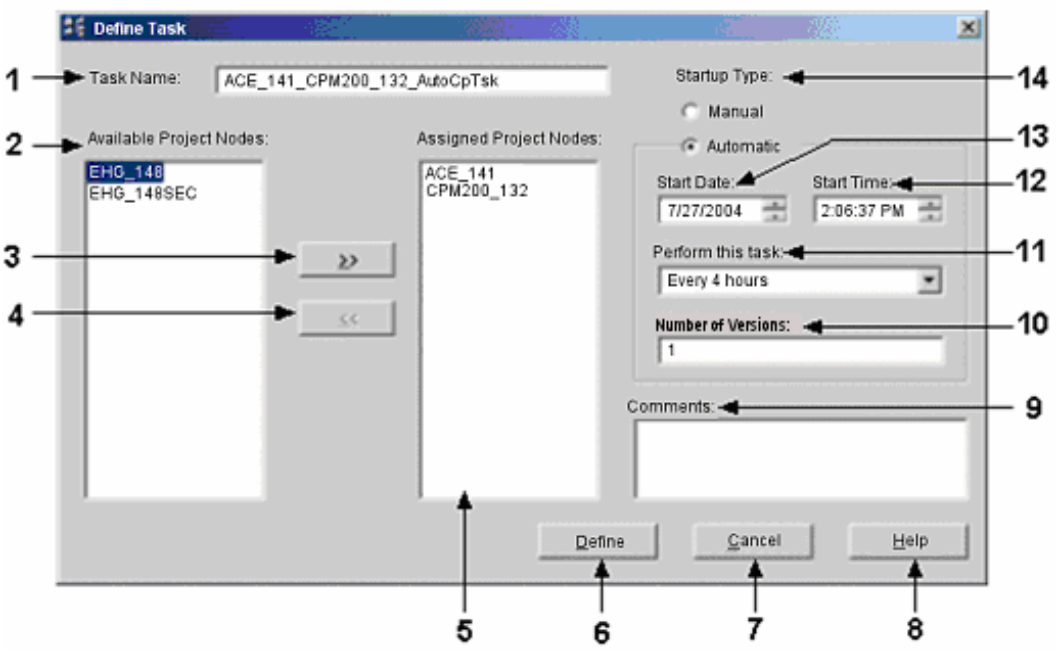
Callout	Function	Description
1	**Task** Tab	Click to view. Contains list box that shows all the scheduled tasks currently configured.

Callout	Function	Description
2	**Current tasks** box	Shows all currently configured tasks with grid that includes: • **Name** of task • **Schedule** set for the task - This is the time interval with the starting time and date as configured in the **Define Task** dialog. Only valid for Automatic Type • **Status** of the task - Can assume Running, Stopped or Scheduled condition. See the following *Checkpoint scheduler state conditions related to stop and start actions* section for more information. • **Type** of task - Can be Automatic or Manual as configured in the **Define Task** dialog. • **Last Run Time** for task - This is the primary server's time when the task's previous/current run was initiated. It is not the time when the task completed, but when the task's run was initiated. • **Next Run Time** scheduled for task - See the following *Definition and computation of next run time* section for more information. • **Creator** of task
3	 **Start** button	When available, click to start selected task. Selecting a Stopped task makes the button available.
4	 **Stop** button	When available, click to stop selected Running or Scheduled task. Selecting a running or Scheduled task makes the button available.
5	**New** button	Click to open the **Define Task** dialog to schedule a new task.
6	**Edit** button	When available, click to open the **Define Task** dialog to edit a selected task. Selecting a Stopped or scheduled task makes the button available.
7	**Delete** button	When available, click to delete the selected task from the schedule and the ERDB. Selecting a Stopped or scheduled task makes the button available.

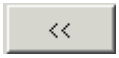
Callout	Function	Description
8	**Refresh** button	Click to refresh contents in **Current Task** box. The Checkpoint Scheduler dialog does not have its status information updated automatically while the display is shown. It must be manually refreshed by either clicking this button or calling up the display again.
9	**Close** button	Click to close the dialog.
10	**Help** button	Click to view associated Help information.
11	**Task by node** tab and **Available** box	Click to view. Tab contains list boxes that show available top-level-hardware-parent entities or nodes and the scheduled tasks that includes the selected node in the **Available** box. Provides a means to filter the display of scheduled tasks by associated node.
12	**Tasks which included the selected node** box	Shows all currently configured tasks that include the selected available node with grid that includes: • **Name** of task • **Schedule** set for the task • **Status** of the task • **Type** of task • **Last Run Time** for task • **Next Run Time** scheduled for task • **Creator** of task
13	**Edit** button	When available, click to open the **Define Task** dialog to edit a selected task. Selecting a stopped task makes the button available.
14	**Refresh** button	Click to refresh contents in the **Tasks, which include the selected node** box.
13	**Close** button	Click to close the dialog.
14	**Help** button	Click to view associated Help information.

Identifying Functions on the Define Task Dialog

The following illustration and callout table, identify the functions that are available on the **Define Task** dialog



Define Task Dialog

Callout	Function	Description
1	**Task Name**	Defines unique name for scheduled task within the server's name space. Default name is the name of the node that is included in the scheduled task. The task name is updated as other nodes are added. Users can also type a desired name in the field, which overrides the default name. See the following *About naming tasks* section for more information. The default name has a suffix of AutoCpTsk, if the task is Automatic type; or a ManCpTsk suffix, if the task is Manual type.
2	**Available Project Nodes** box	Shows all currently configured nodes in the **Project** tab.
3	 **Add** button	When available, click to add selected available node to **Assigned Project Nodes** box. Selecting a node in the **Available Project Nodes** box makes the button available. You can use common Window's shortcut keys to make multiple selections.
4	 **Remove** button	When available, click to remove selected node from the **Assigned Project Nodes** box. Selecting a node in the **Assigned Project Nodes** box makes the button available. You can use common Window's shortcut keys to make multiple selections
5	**Assigned Project Nodes** box	Shows all nodes currently assigned to the scheduled task. Use the **Add** and **Remove** buttons to edit selections.
6	**Define** button	When available, click to create the named scheduled task and have it added to the **Current tasks** box on the **Tasks** tab of the **Checkpoint Scheduler** dialog. Entering a Task Name and adding at least one available node to the **Assigned Project Nodes** box makes the button available.
7	**Cancel** button	Click to close the dialog and end the create or edit schedule task operation without saving changes.
8	**Help** button	Click to view associated Help information.

Callout	Function	Description
9	**Comments**	Type any comment you want associated with this task in the box. This could be information to help identify the file for a restore operation. This comment is stored in each checkpoint file produced by this task. The comment will appear in the Checkpoint Restore User interface when the file is displayed for restore selection.
10	**Number of versions**	Defines the maximum number of files that will be created per node in this task. Once this maximum number is reached, the older versions are overwritten with the newer checkpoint files. The default number of files is set to 5. The default is set to 1 in case of a Qualification and Version Control System (QVCS) enabled system license and cannot be changed. This field is only available when **Startup Type** selection is **Automatic**.
11	**Perform this task**	Click arrow button to select desired task interval time from the list: • Every 2 hours • Every 4 hours • Every 8 hours • Every Day • Every 3 days • Weekly This field is only available when **Startup Type** selection is **Automatic**. The default selection is **Every 4 hours**.
12	**Start Time**	Click the up and down arrow buttons to set the desired time for the task to start. This field is only available when **Startup Type** selection is **Automatic**. You can also key in the values in the selected time field.
13	**Start Date**	Click the up and down arrow buttons to set the desired date when the task is to start. This field is only available when **Startup Type** selection is **Automatic**. You can also key in values in the selected date field.

Callout	Function	Description
14	**Startup Type**	Select **Manual** or **Automatic** as the startup type for the task. You cannot schedule a **Manual** type task. To run a **Manual** type task, select the task in the **Checkpoint Scheduler** dialog and click the Start button. If a **Manual** task is chosen, the following boxes are unavailable and **cannot** be configured: **Start Date**, **Start Time**, **Perform This Task**, and **Number Of Versions**.

Scheduling Checkpoint Tasks

This section provides a procedure for using the Checkpoint User Interface to schedule tasks to automatically checkpoint selected hardware entities at predetermined intervals.

Prerequisites:

- You have logged on with a security level that permits you to schedule checkpoint tasks through Control Builder.
- You have configured control strategies in Control Builder.
- The Windows Task Scheduler service is running (Started).



CAUTION

Never make Checkpoint schedule changes through the Windows Task Scheduler service. The Checkpoint Service uses periodic timing triggers through Windows NT Tasks that it has configured. Checkpoint Service assumes that it only is creating/changing/deleting these triggers. You must never modify these related NT Tasks or stop the Windows NT Scheduler Service.

Considerations:

- If no tasks are currently configured, selecting the **Schedule Checkpoint Tasks** command calls up the **Define Task** dialog box instead of the **Checkpoint Scheduler** window. The status of each task is updated when the **Checkpoint Scheduler** window is opened. You can also click the **Refresh** button to update the status of each task in the **Checkpoint Scheduler** window.
- You can run a manual task by clicking the  **Start** button on the **Checkpoint Scheduler** window.
- All applicable nodes in the **Project** tab of Control Builder are listed in the **Available Project Nodes** box on the **Define Task** dialog. A scheduled task is run according to the configuration settings that you made through the **Define Task** dialog whether or not a node within the task has been loaded. When that task runs, the node that was not loaded will not be saved. Checkpoint save/restore is only applicable for nodes that appear in the **Monitoring** tab.
- Once you schedule a task, up to five nodes in the task are checkpointed concurrently. See **Deferral and Resumption Of Node Saves** for information related to checkpoint save deferral and resumption. When a node's checkpoint save

completes within the given task, that checkpoint file is created on the Server for that node.

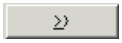
- For **Automatic** checkpoint save, the initial save of the checkpoint file for a node with no previous versions is written to the disk regardless of the required free space. If this save violates the required free space, an event is generated. If there is not enough disk space available to store this initial checkpoint file, an event is generated and the checkpoint file is saved.
- For **Automatic** checkpoint save, if there are previous versions of the checkpoint file for a node and the required free space is violated, the oldest checkpoint file version is deleted and the new version is copied to the disk. Before saving the new file, checks are made to ensure that the space regained by deleting the old checkpoint file is enough to store the new checkpoint file. If the regained space is inadequate, the next older versions are deleted until there is enough disk space available. This reduces the number of checkpoint file versions available for the given node. At least one checkpoint file is maintained for each node in the given automatic task. For example, if a given node is in three automatic tasks and all saves are causing free space violations, attempts will be made to save three checkpoint files for that node (one for each task)
- You can use any compatible checkpoint files to restore a node to its previous configuration independent of ERDB presence.
- You can convert a **Manual** type scheduled task to an **Automatic** one by editing the **Startup Type** for the task on the **Define Task** dialog or vice versa - change an **Automatic** type to a **Manual** one.
- The initial task status for a Manual type task is Stopped
- The initial task status for an Automatic type task is Scheduled
- The following table summarizes the interaction between the **Start** and **Stop** buttons for a given task status and type.

Task Status	Task Type	Start Button	Stop Button
Running	Automatic	Disabled	Enabled
Running	Manual	Disabled	Enabled
Stopping	Automatic	Disabled	Disabled
Stopping	Manual	Disabled	Disabled

Task Status	Task Type	Start Button	Stop Button
Stopped	Automatic	Enabled	Disabled
Stopped	Manual	Enabled	Disabled
Scheduled	Automatic	Disabled	Enabled
Scheduled	Manual	Illegal Condition –Should never happen.	

The following procedure assumes tasks were not previously configured and outlines the steps typically involved with scheduling a task to run automatically.

To create and schedule an automatic checkpoint task

Step	Action
1	On the **Controller** menu, click **Checkpoint** > **Schedule Checkpoint Tasks**.
2	• If the **Define Task** dialog box appears, go to the next step • If the **Checkpoint Scheduler** window appears, click the **New** button to call up the **Define Task** dialog box.
3	In the **Task Name** field, you can choose to type in a desired name for this task that is ensured to be unique to the server's name space. The default name consists of a series of node names that are assigned to the task.
4	In the **Available Project Nodes** box, select one or more nodes you want to include in the task.
5	Click the  Add button to include the node selections in the **Assigned Project Nodes** box. Note that series of node names in the **Assigned Project Nodes** box are inserted as the default name in the **Task Name** field.
6	Under **Startup Type**, click **Automatic**.
7	In the **Start Date** field, click  up or down arrow button to set desired date that task is to start.
8	In the **Start Time** field, click the  up or down arrow button to set desired time of day that task is to start.
9	In the **Perform this task** field, Click the  down arrow button and select the desired time interval for the task from the list.

Step	Action
10	In the **Number of Versions** field, type the maximum number of files you want created for each node in this task. The default number is 5 unless you are running a licensed QVCS application on your system, then the default number is 1. The maximum number of versions allowed is 100.
11	In the **Comments** field, type any desired message that will help you select the checkpoint file for this task during a restore operation.
12	Click the **Define** button to create the task, add it to the **Tasks** tab on **Checkpoint Scheduler** dialog, and close the **Define Task** dialog.
13	This completes the procedure.

About naming tasks

The default naming action continues for a given task until you manually enter a task name. For example, if you select a node in the **Available Project Node** list box and transfer the node to the **Assigned Project Node** list box, the node's task name character string is concatenated to the present task name. If you transfer the node back from the **Assigned Project Node** list box to the **Available Project Node** list box, the string of characters associated with the node's name are removed from the existing task name.

When you enter a string into the **Task Name** box, the default naming action for that given task is permanently disabled. This applies even if you would enter what would be the default name.

When you click the **Define** button, the system ensures that the task name is unique within an Experion server name space among all tagged block names. Also, it ensures the task name does not exceed 40 characters and does not have illegal characters as defined by the Experion tag naming convention. If a task name does not comply, an error message prompts you to take corrective action before proceeding.

Checkpoint scheduler state conditions related to stop and start actions

The following table summarizes some scheduler state conditions for given user initiated start and stop actions.

Task Status	Task Type	User Action	Next Task Status	What happens
Running	Automatic	Stop	Stopping	Current Save in progress accessing the Nodes. This occurs until the node saves are completed. However, the task gets unscheduled while saves are completing, Next Run time is set to blank
Running	Manual	Stop	Stopping	Current save in progress accessing the nodes. This occurs until the node saves are completed.
Stopping	Automatic and/or Manual	None possible	Stopped	Task is in process of stopping. When all associated nodes have saves completed then task state progresses to stopped. Prior to taking any other task action, the user must wait until checkpoint service fully stops the task. This gets indicated by the “Stopped” state.
Stopped	Automatic	Start	Scheduled	Task is scheduled with the original start time and date when the task was defined and is scheduled to run at the next interval. The interval information is obtained from “define task” as well.
Stopped	Manual	Start	Running	Task runs with the start time and date when the **Start** button was pressed.
Scheduled	Automatic	Stop	Stopped	Task is unscheduled

Task overrun condition

A Task can overrun if it does not complete before its next interval. In this case, all next scheduled intervals are skipped until the current interval completes. After completing the current interval, an event is journaled showing the normal completion of the task. However, no additional event gets journaled to signal that the next scheduled running of the task has overrun.

The task is not re-run immediately because of the overrun. The task only runs again at its next interval. It is the user's responsibility to adjust their schedules to avoid too many delayed/overrun occurrences.

Deferral and Resumption Of Node Saves

In R300.1, to throttle checkpoint service's consumption of the primary server's CPU processing resource, the Checkpoint Service uses a mechanism for deferring and resuming saves. The checkpoint service limits the number of simultaneous node (entity) saves to 5 (five). This applies to all the saves that have been scheduled or demanded in that instance of time. This means that for all the manual saves (from manual on-demand and manual tasks) and automatic task saves that have been engaged to be run only 5 node entities can be placed into active saving execution threads. For any additional nodes that need to be saved, those save request are deferred and placed into a FIFO queue. When an active save completes, the first entity on this deferred FIFO save queue gets removed and has the freed save thread now assigned to work for saving that entity. Save Deferral Events get journaled for all entities that have their saves deferred. A *resumption save* event gets generated for each entity, when its save is resumed as previously explained.

Definition and computation of task's elapsed run time

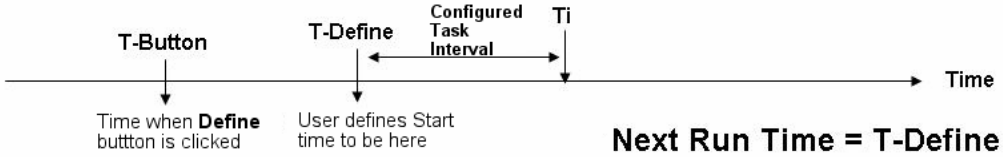
A task's elapsed run time is not provided as a parameter on the task. If you need to determine this, you must view and search the Event Summary for the task's start and completed events applying to the given task in question. The task's elapsed run time will be the difference from the time that this completed event was journaled from when the task's corresponding start event was journaled. This elapsed time may vary from run to run depending on what other checkpoint saves are going on and their affect on deferrals and resumptions of the node saves within the task being considered.

Definition and computation of next run time

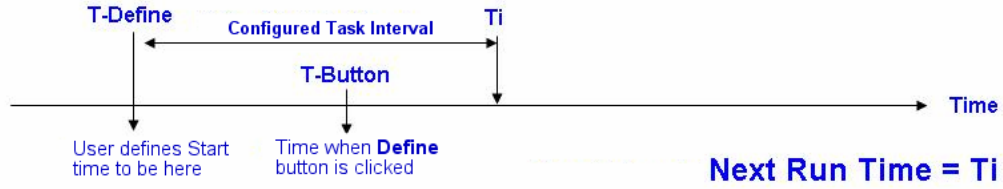
The next run time is defined as the time when the Automatic task will be started next based on the schedule. It does not apply to Manual type tasks. When an Automatic task is stopped, it is taken out of the schedule and therefore there is no next run time.

Checkpoint uses the Windows Scheduler Service to make all next run time computations. Even if you define an automatic task with a Start Date and Start Time in the past or before the present time when the actual **Define** button is clicked, the Windows Scheduler Service computes the next run time. In this case, the next run time is offset by the interval configured in the **Perform This Task** box that is the first time mark that occurs after the present time or when the **Define** button is clicked. The following time line examples illustrate this.

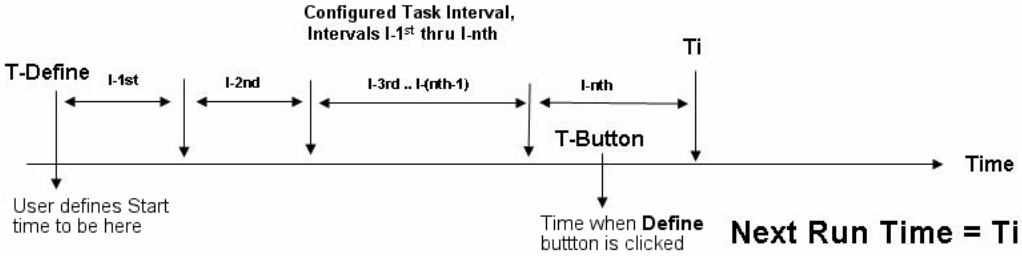
Example 1: Next run time is computed based on a defined start date and time that will still be in the future after the **Define** button is clicked.



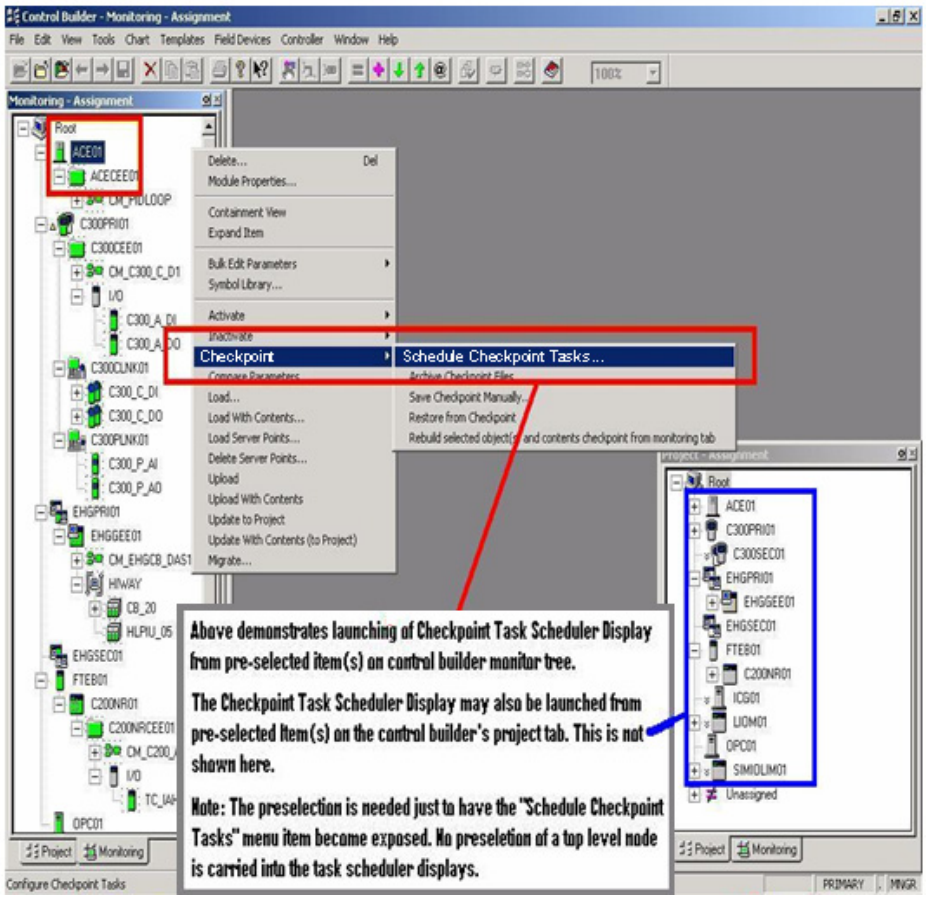
Example 2: Next run time is computed based on a defined start date and time that is in the past by less than one configured Task interval after the **Define** button is clicked.



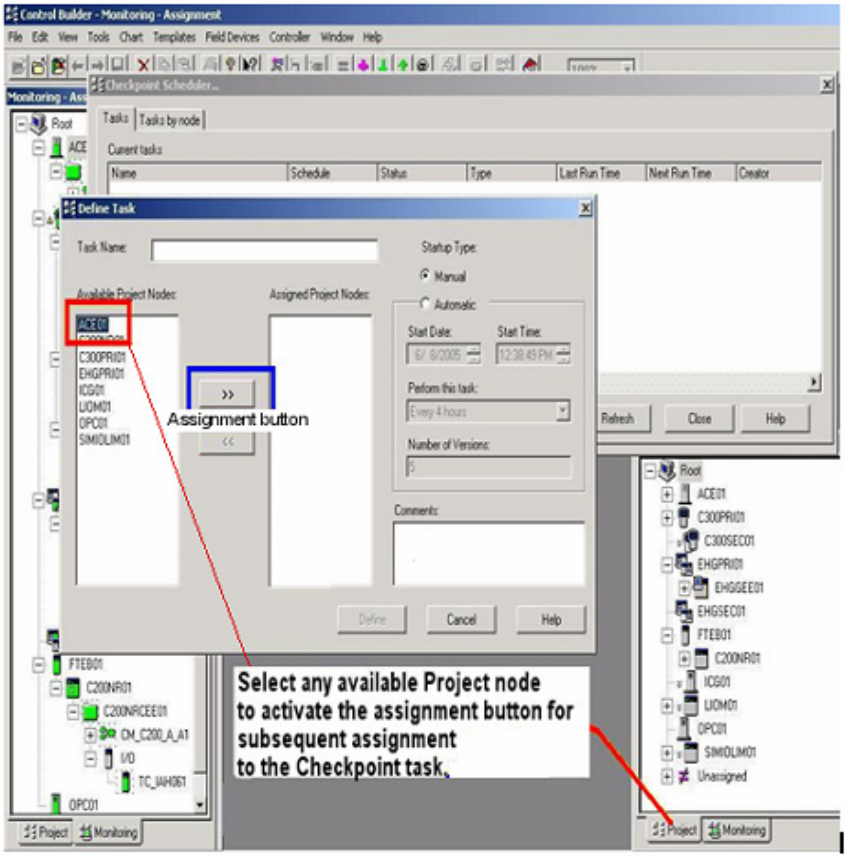
Example 3: Next run time is computed based on a defined start date and time that is in the past by more than one configured Task interval after the **Define** button is clicked.



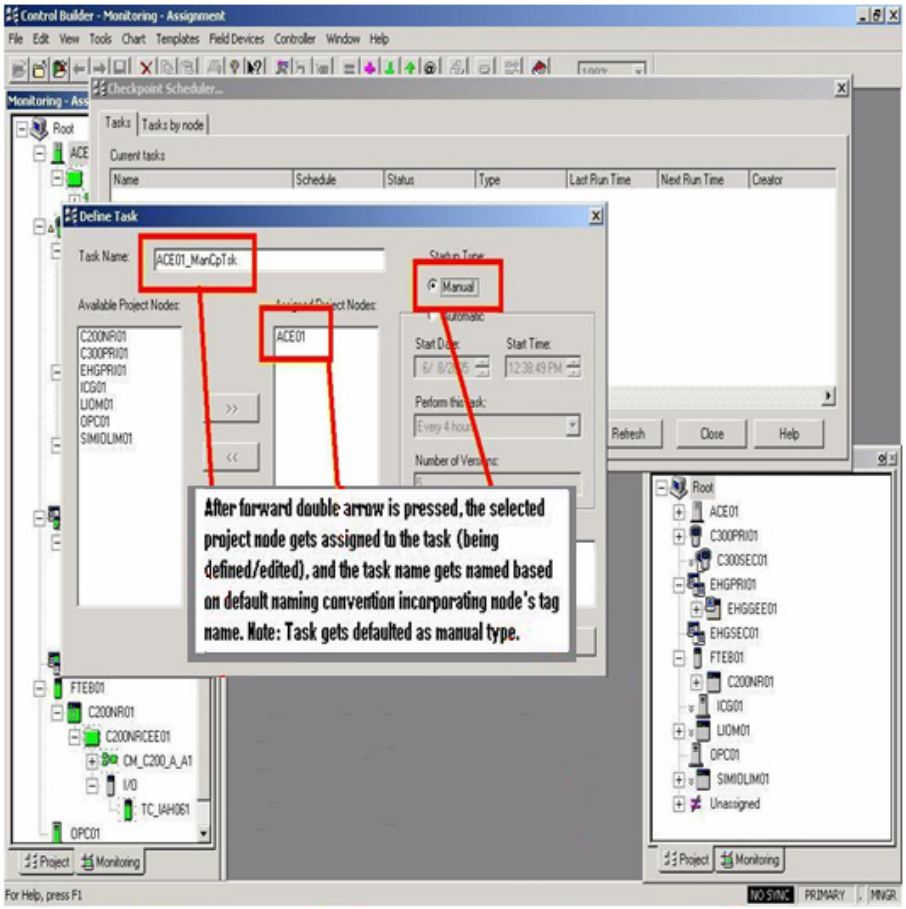
Checkpoint scheduler graphical reference



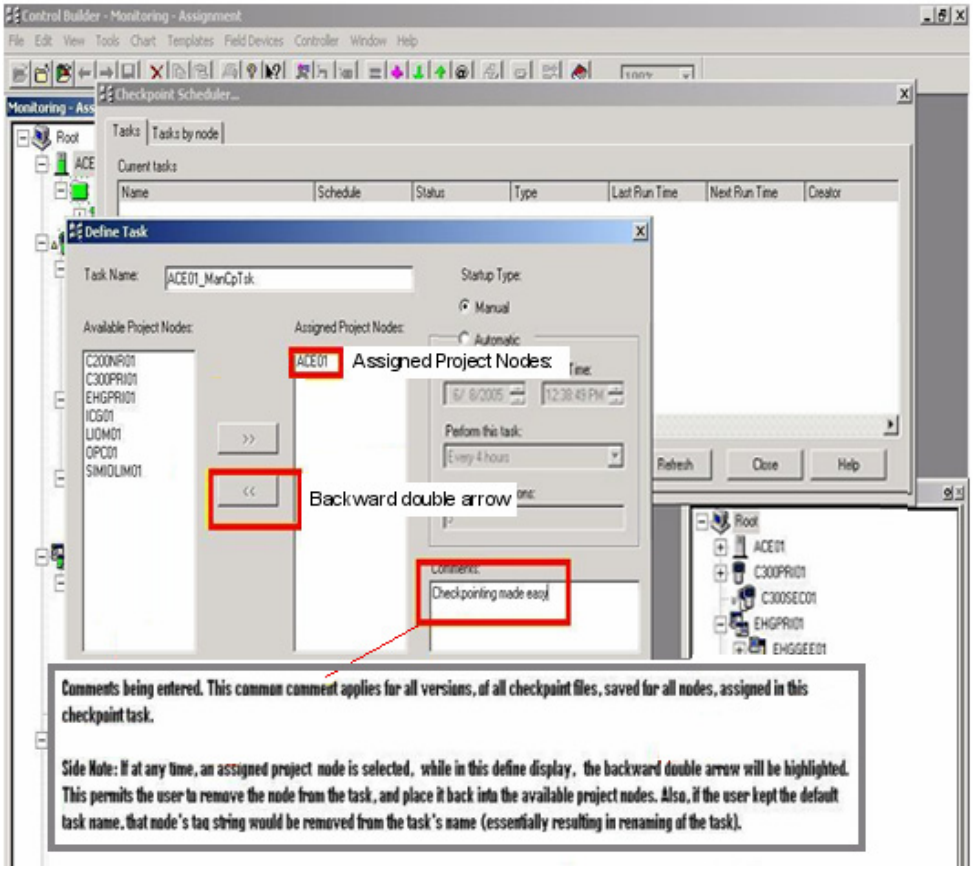
Launching Checkpoint Scheduler dialog



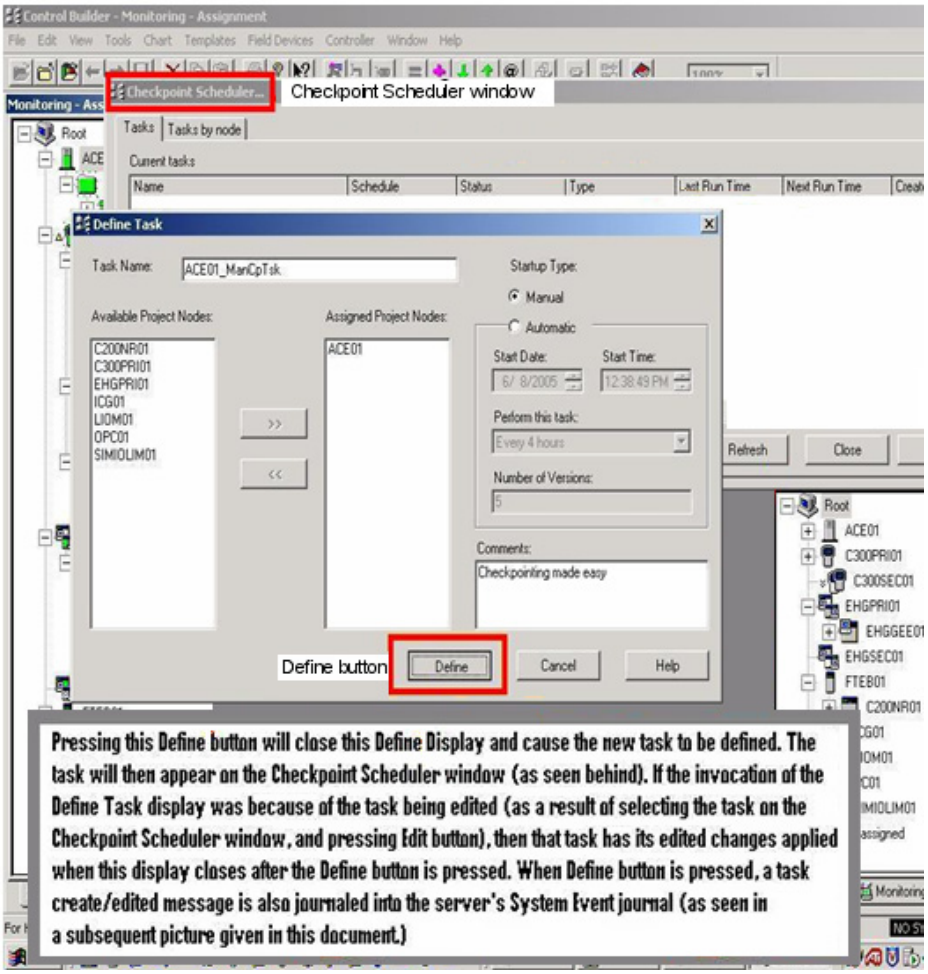
Selecting a Project node in the Define Task dialog



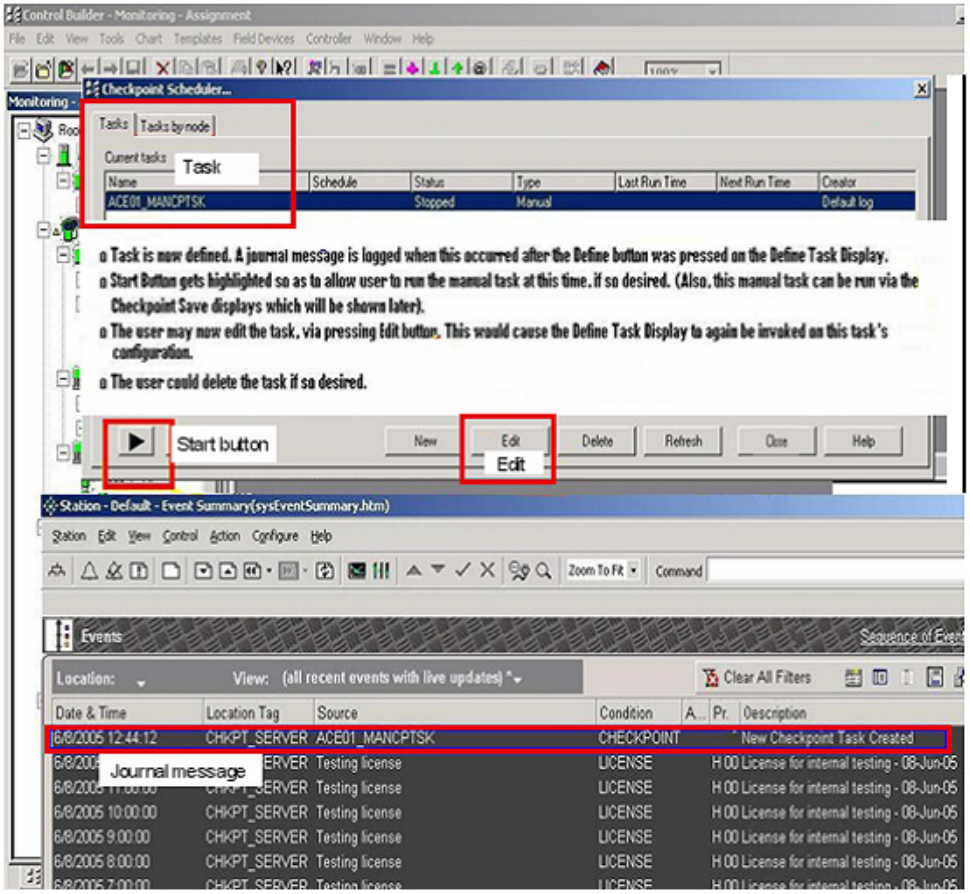
Added/assigned node gets default Task Name created



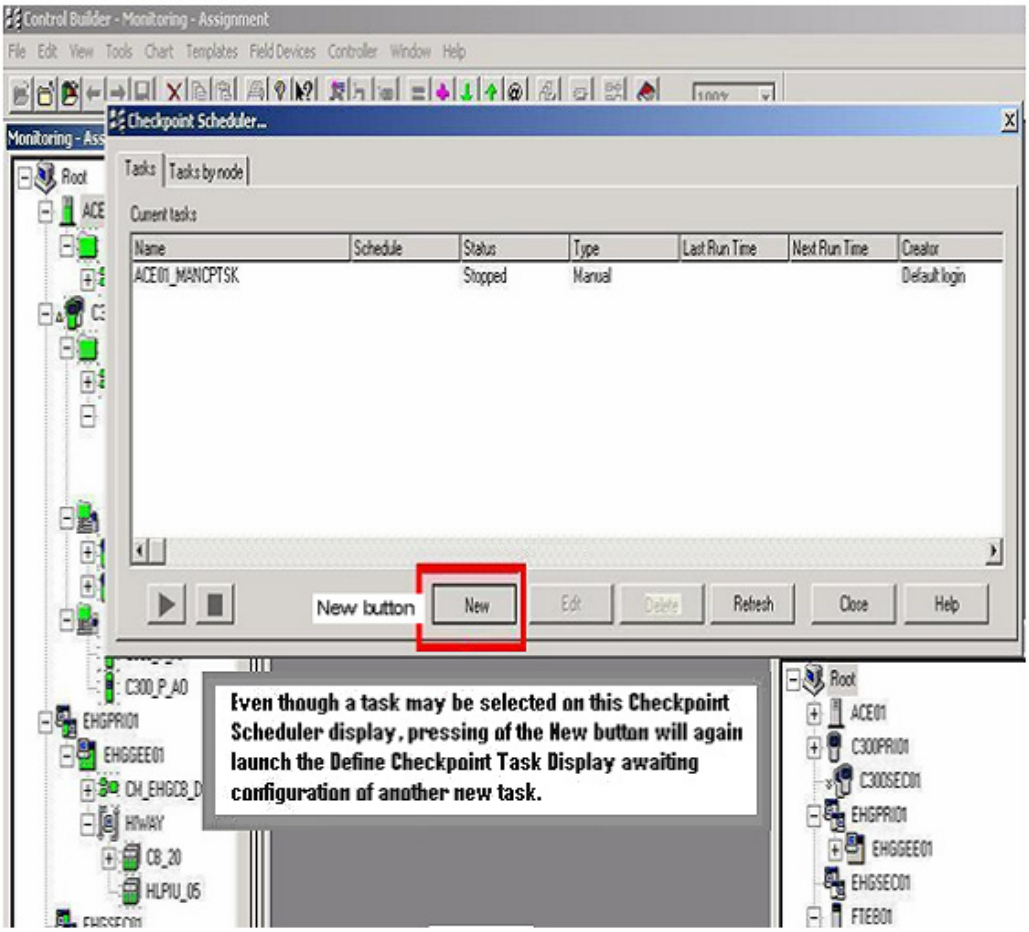
Adding comments and how to remove assigned node



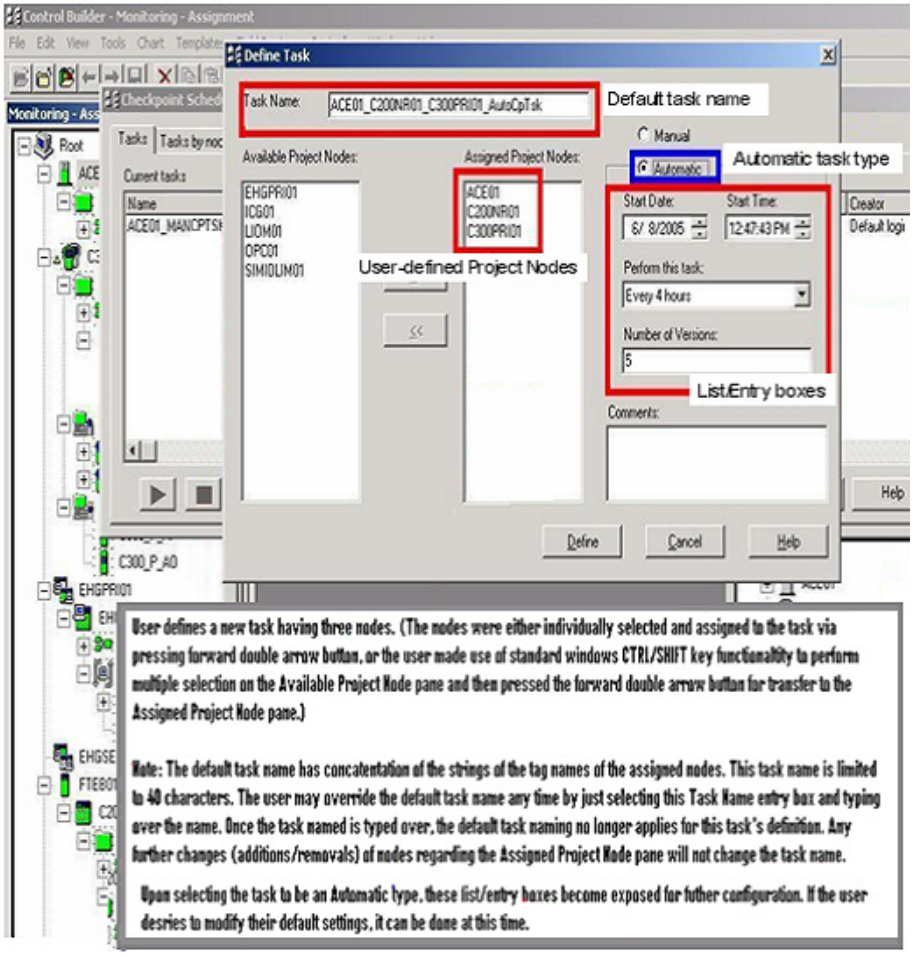
Click Define button to create configured task checkpoint schedule and close Define Task dialog



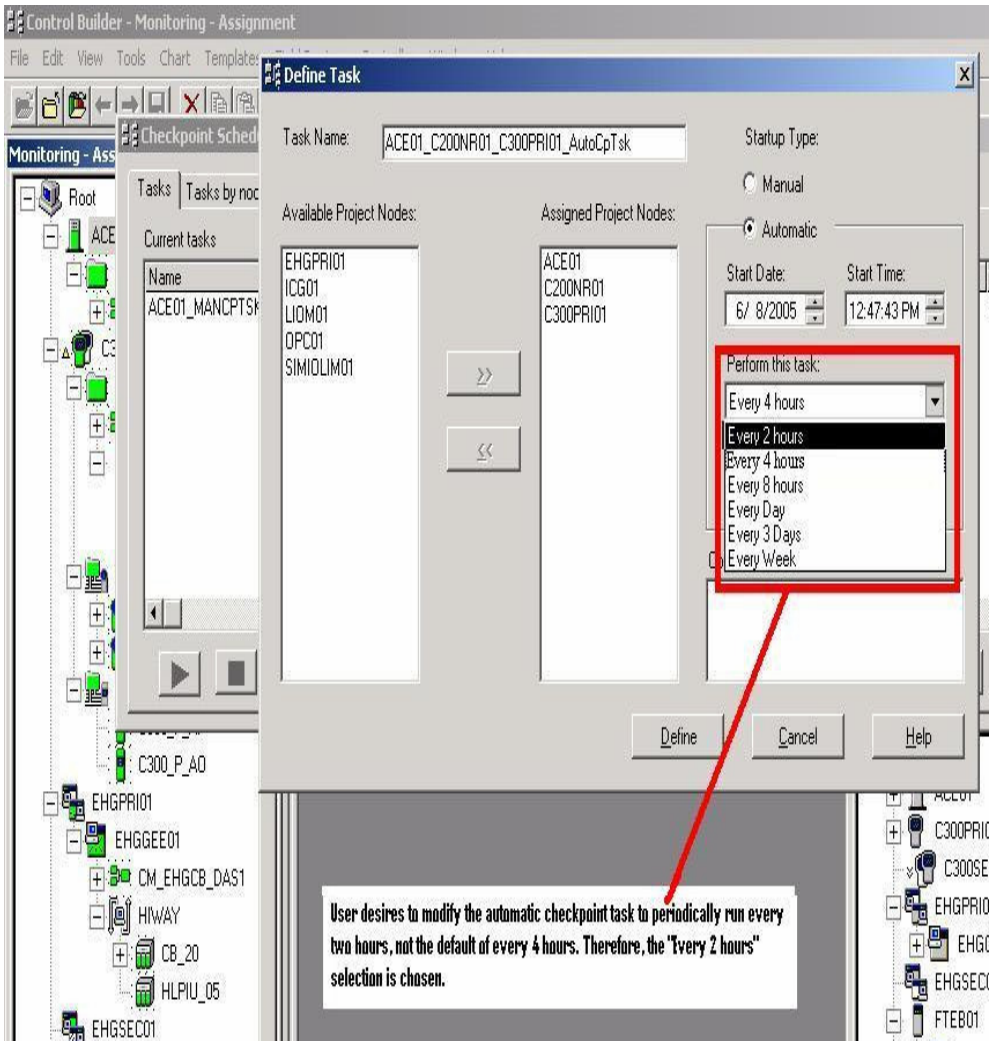
Task is defined in Checkpoint Scheduler and event is journaled in Station Event Summary display



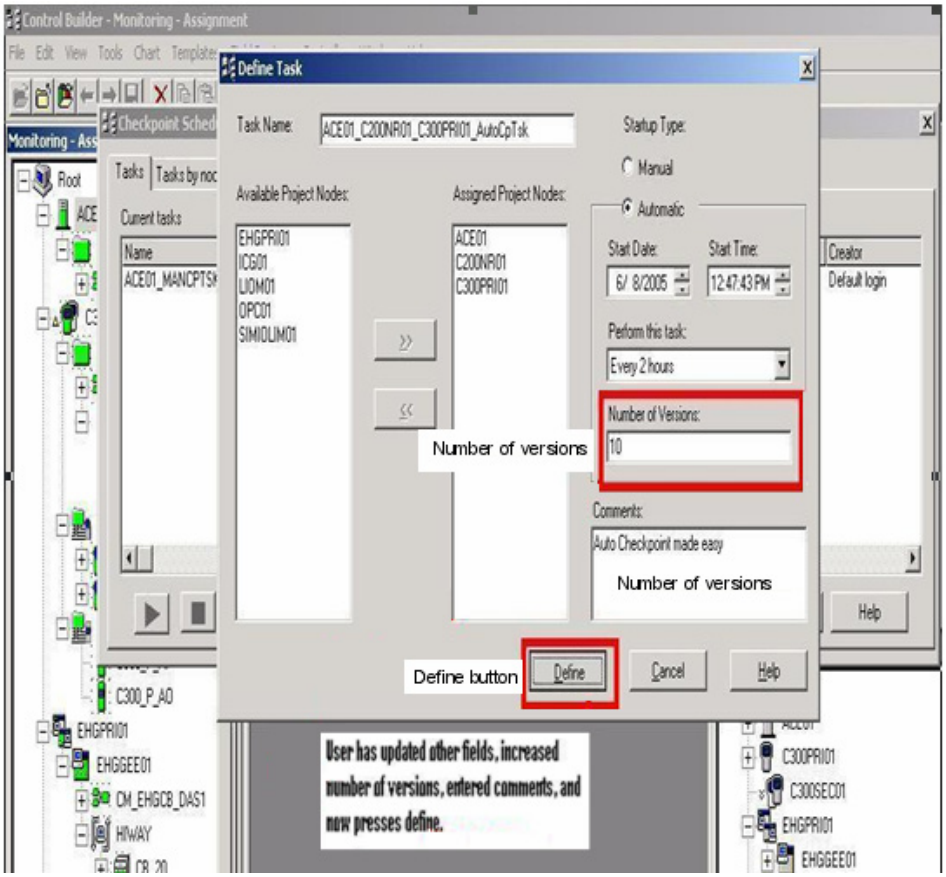
Click New button to launch Define Task dialog to configure another checkpoint schedule task



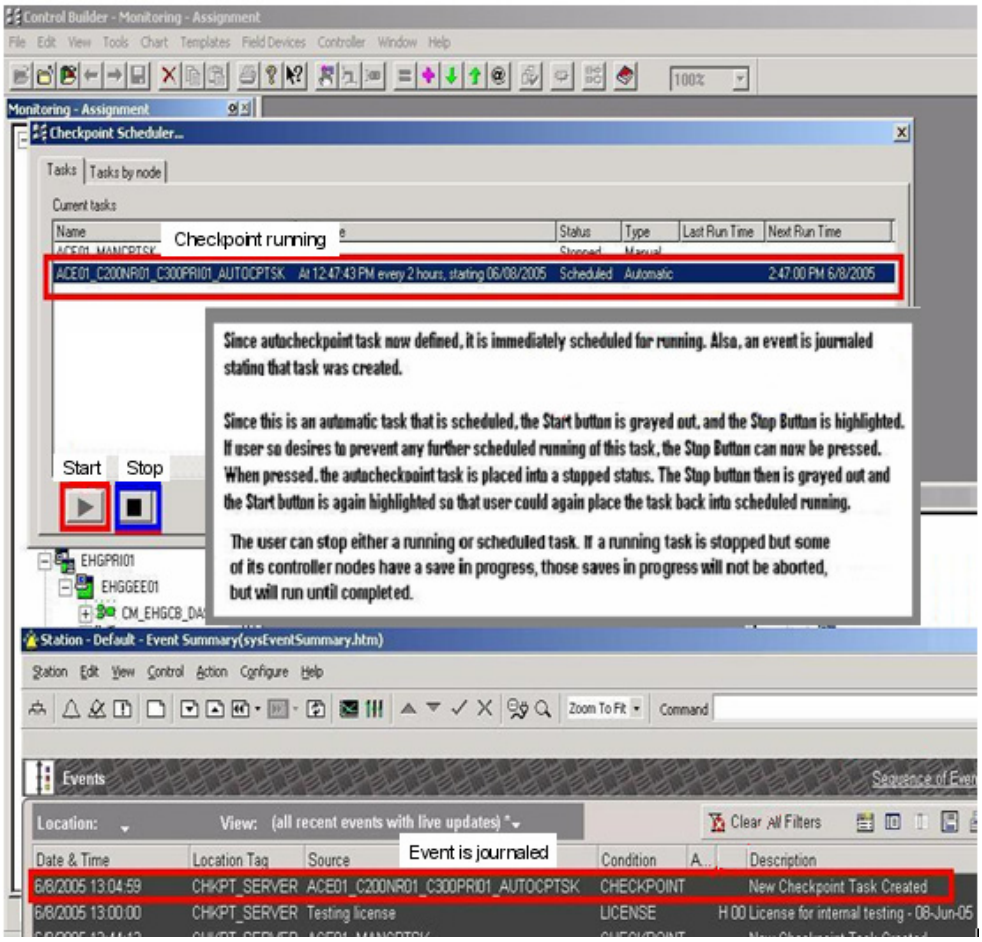
Configuring Automatic type task checkpoint schedule for three assigned Project nodes



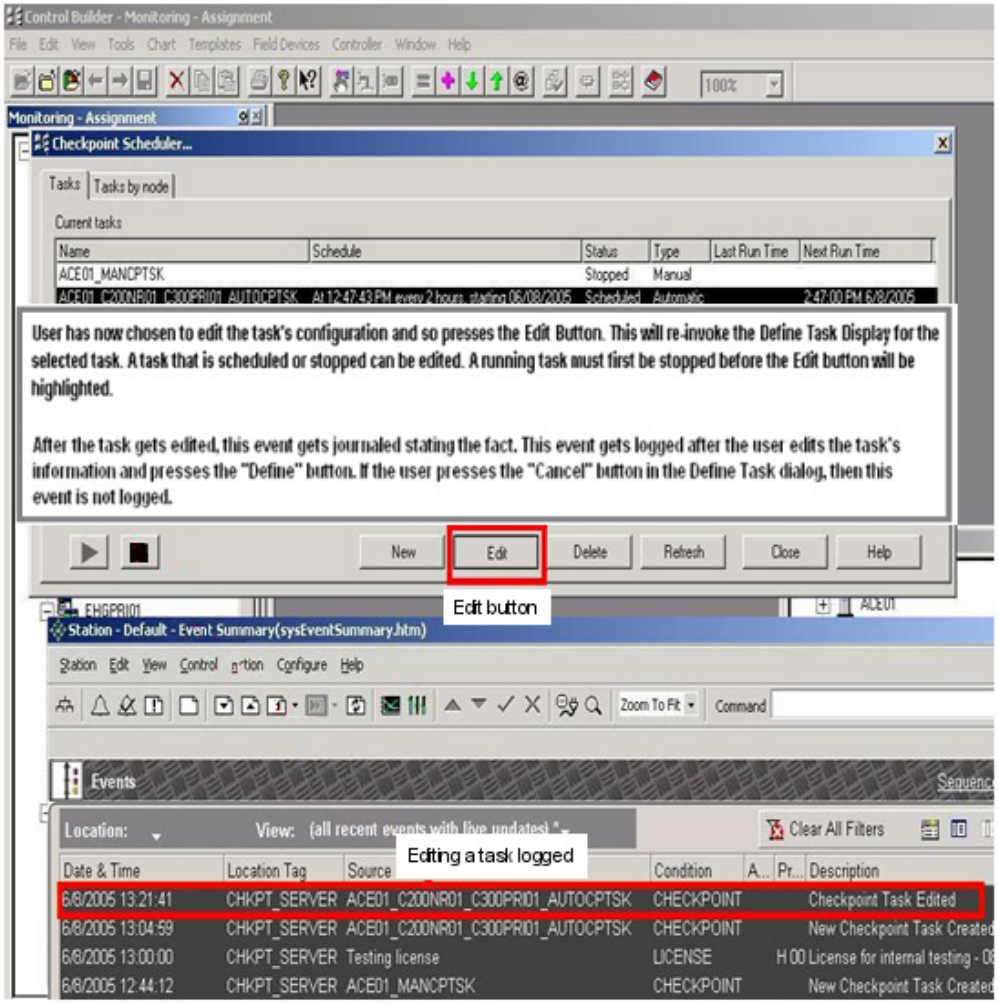
Changing Task time interval from Every 4 hours to Every 2 hours



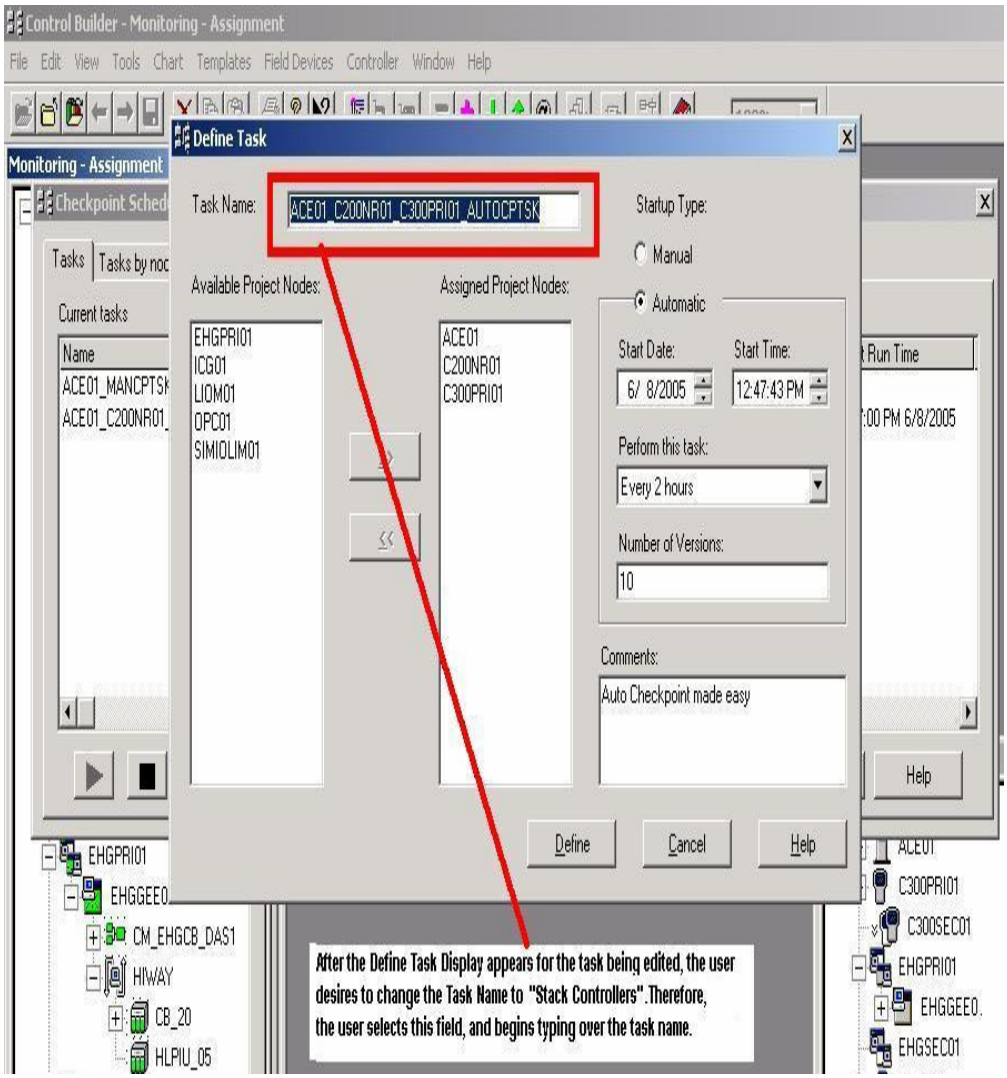
Editing configuration fields and clicking Define button to save changes



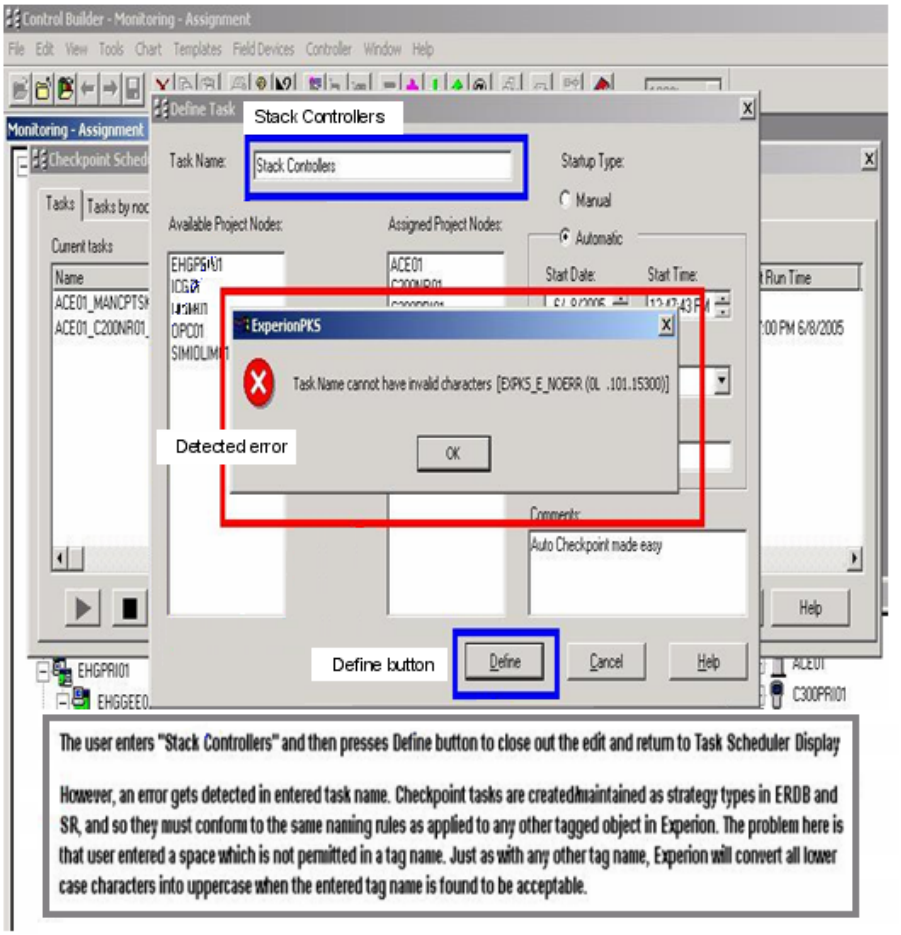
Automatic checkpoint task is now defined and scheduled to run



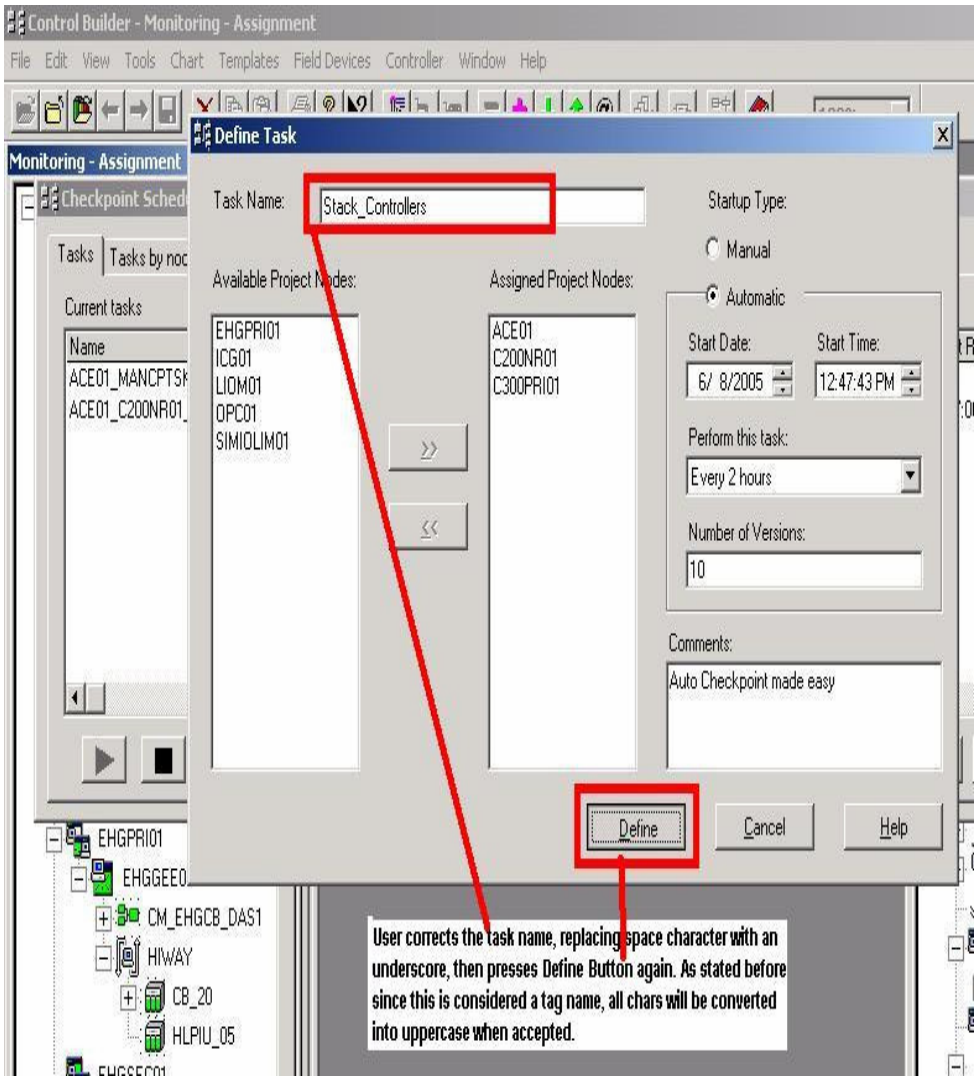
Making edits to selected checkpoint schedule task



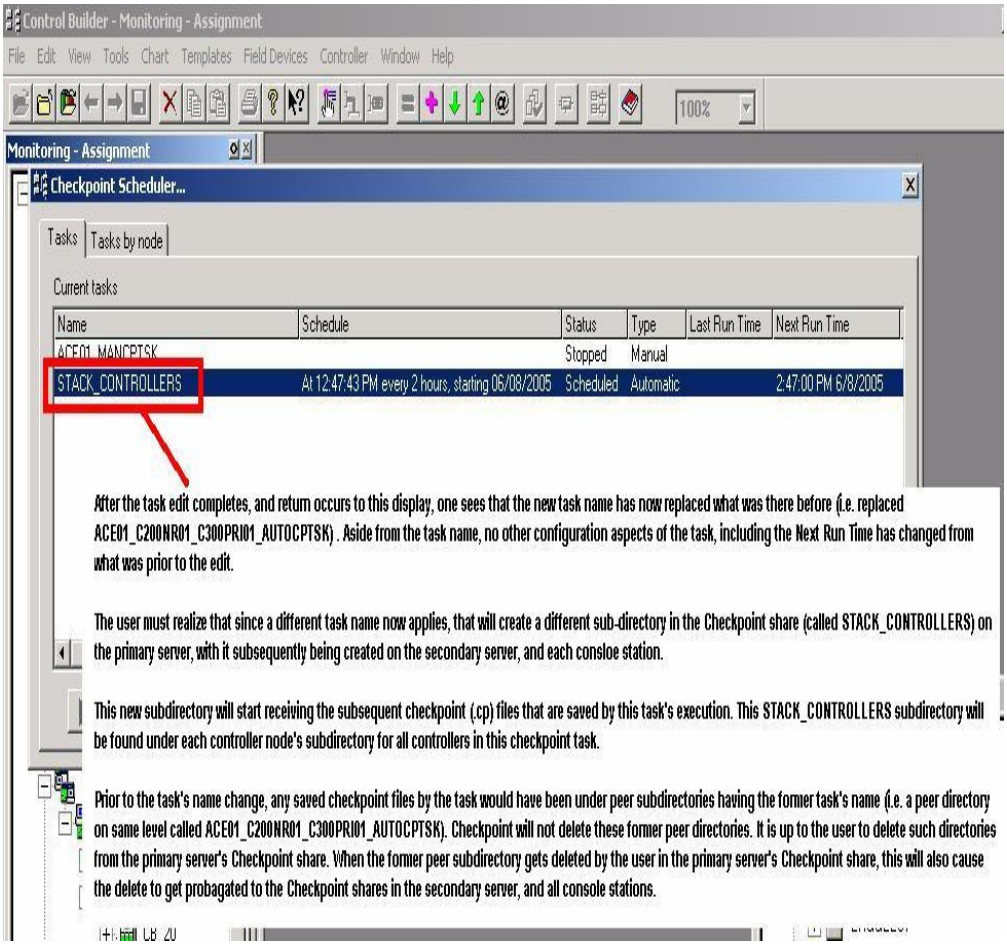
Selecting current task name for editing



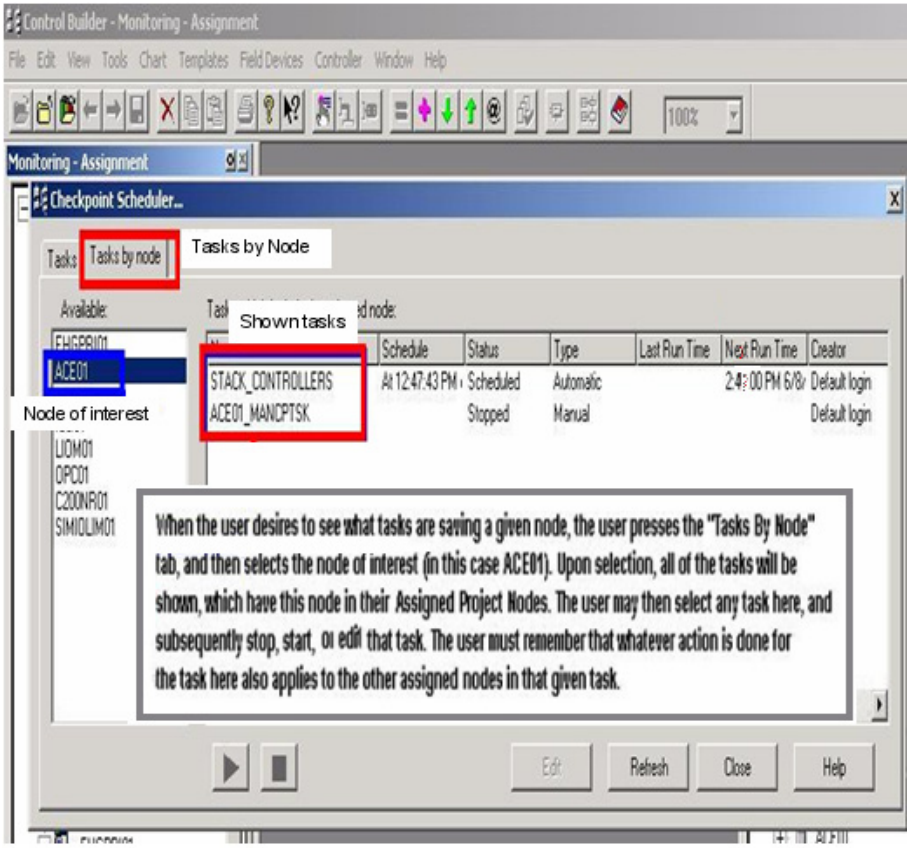
Entered Task Name edit includes invalid character



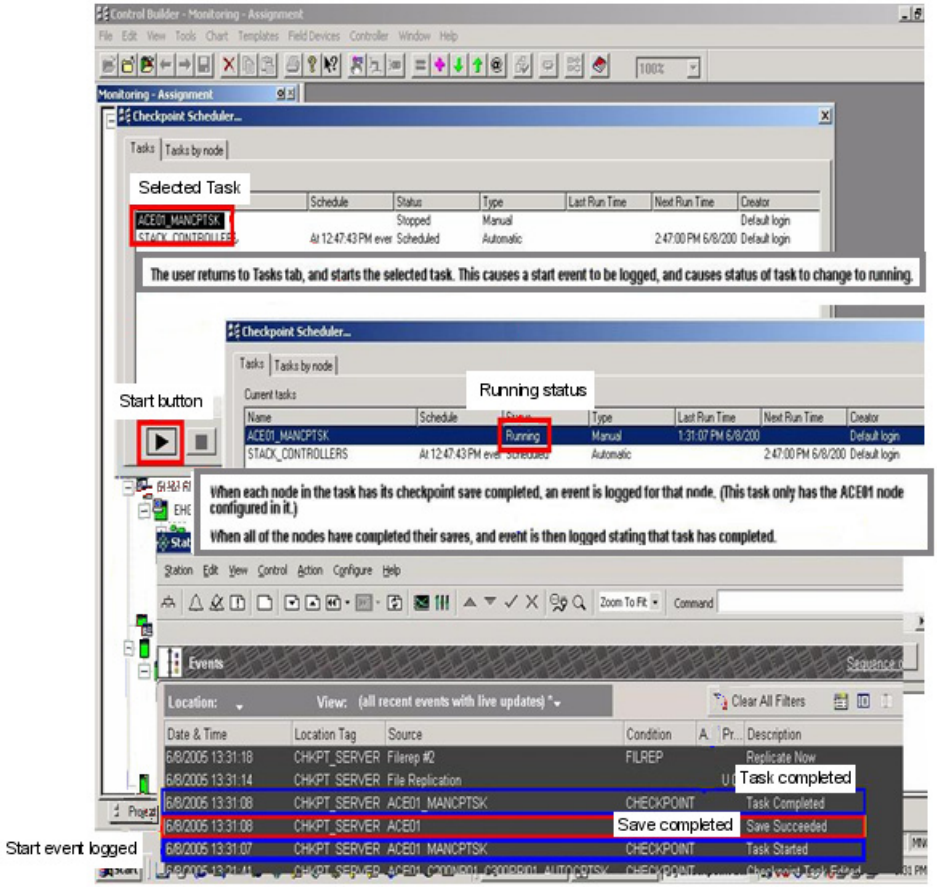
Correct invalid character error by replacing space with valid underscore character and click Define to save changes



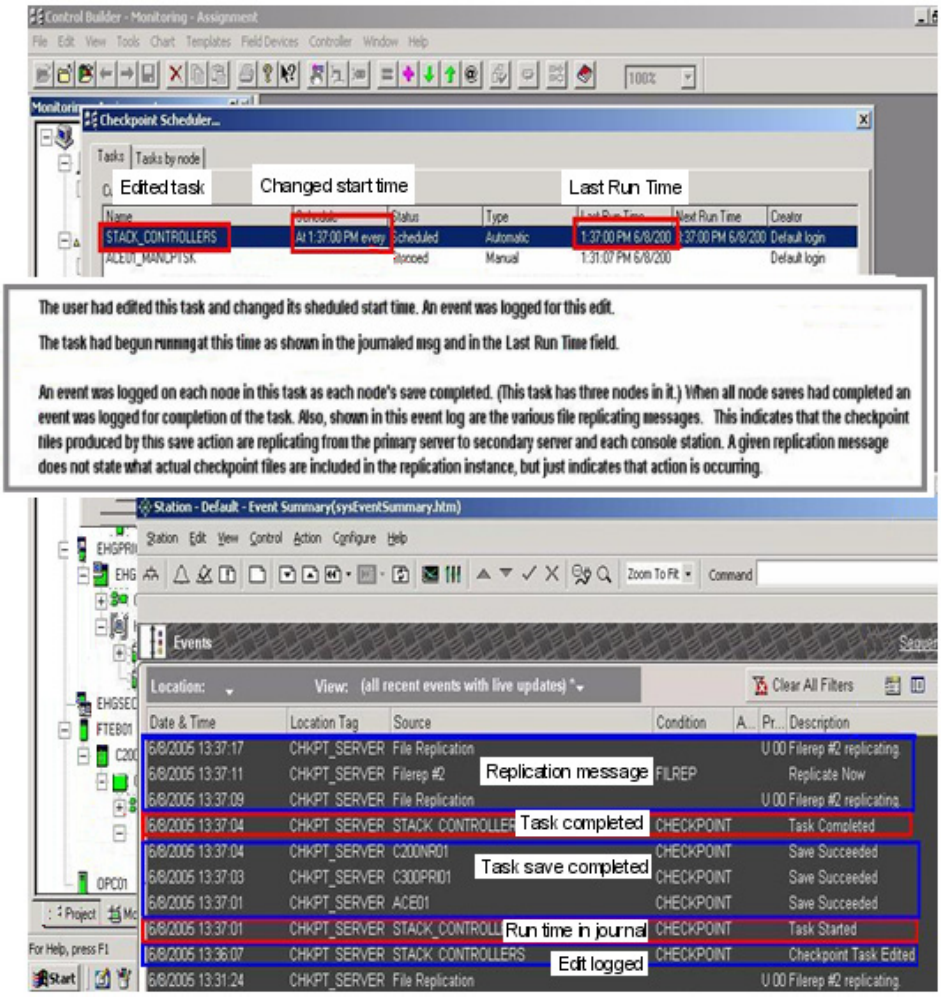
Edited Task Name now appears as Name for task in Checkpoint Scheduler



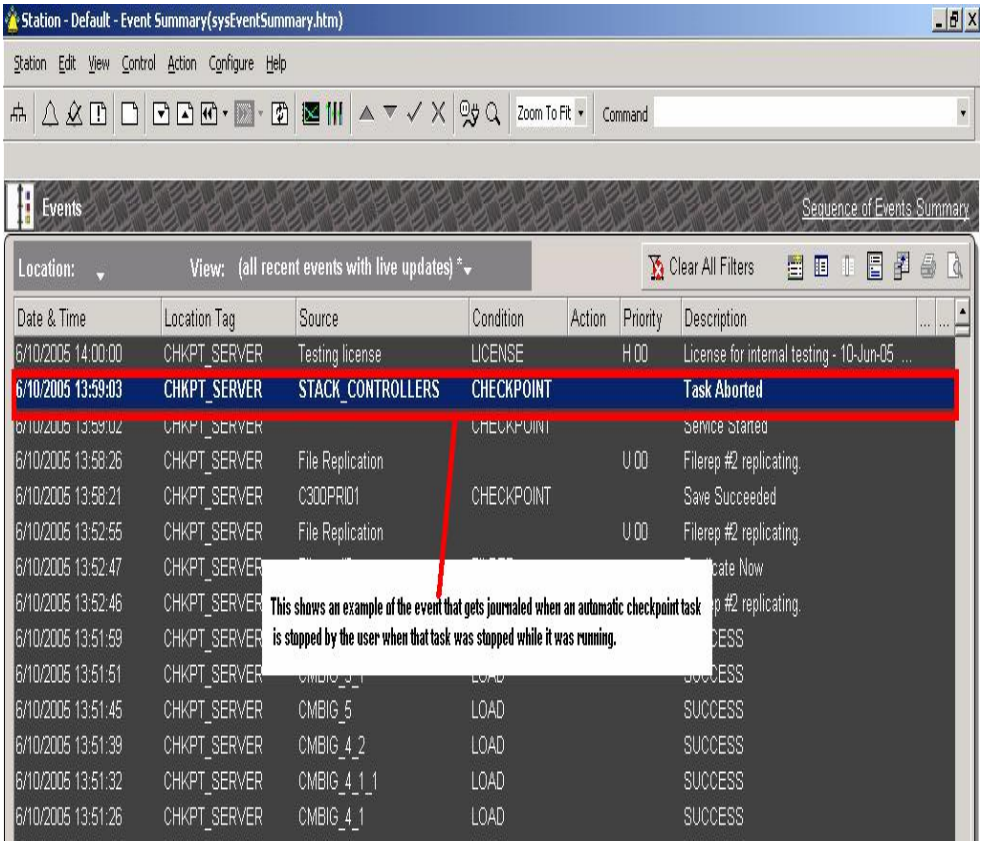
Click Task by node tab to view all tasks associated with a given node



Click Tasks tab, select Manual checkpoint schedule task, click Start button, monitor status on tab, and view events journaled in Station Event Summary display



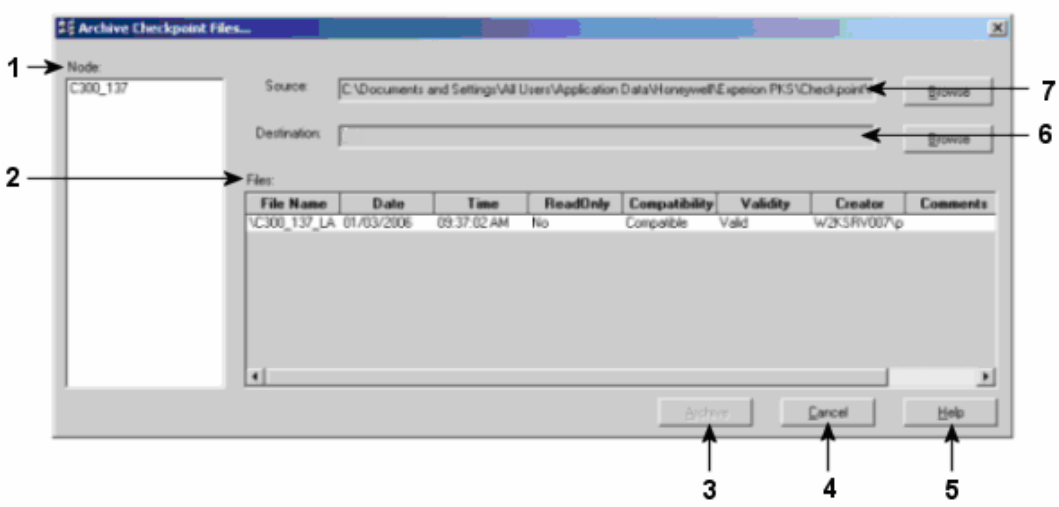
Check status of automatic checkpoint task on Tasks tab of Checkpoint Scheduler and through events journaled in Stations Event Summary display



Station Even Summary display shows Task Aborted event journal for automatic checkpoint task stopped by user

Identifying Functions on the Archive Checkpoint Files Dialog

The following illustration and callout table identify the functions that are available on the **Archive Checkpoint Files** dialog.



Archive Checkpoint Files Dialog

Callout	Function	Description
1	**Node** box	Show all loaded Project nodes that appear in the **Monitoring** tab.

Callout	Function	Description
2	**Files** box	Shows all saved checkpoint files associated with the selected node in the designated source location with a grid that includes: • **File Name** is the Name of the checkpoint file. • **Date** file was last saved. • **Time** file was last saved • **ReadOnly** marked as Read Only by the user, internally file attribute changed to read only by checkpoint application. • **Compatibility** shows logical integrity of the checkpoint file. Indication is provided on a checkpoint file to inform the checkpoint function when the file is permitted for checkpoint restore. See the following *Checkpoint file attributes* section for more information. • **Validity** shows the physical integrity of the checkpoint file. See the following *Checkpoint file attributes* section for more information. • **Creator** is the user's login account name • **Comments** composed by task creator You can change the viewing order of data from ascending to descending or vice versa by clicking the cursor in a column heading in the table grid on the dialog.
3	**Archive** button	When available, click to initiate store of selected files to the **Destination** location. Prompt message will ask if you want to mark files as ReadOnly. Click the **Yes** button to mark the files as ReadOnly. Selecting one or more files makes the button available.
4	**Cancel** button	Click to close the dialog and end the current session.
5	**Help** button	Click to view associated Help information.

Callout	Function	Description
6	**Destination**	Click the **Browse** button to navigate to the directory location where the files are to be archived or accept the default location. The last selected **Destination** directory is retained as the default directory for the next archive session. The **Source** and **Destination** file locations are different by default. The Archive function does not permit you to have the same destination path as the source path
7	**Source**	Click the **Browse** button to navigate to another directory that includes saved checkpoint files. The **Source** and **Destination** file locations are different by default.

Archiving Checkpoint Files

This section provides a procedure for using the Checkpoint User Interface to archive checkpoint files to another file location or medium for safekeeping until they are needed for a possible restoration operation. This lets you free up the on-line checkpoint storage without losing those checkpoint files that may have tuning and other information that you want to retain but do not need to access immediately.

Prerequisites:

- You have logged on with a security level that permits you to archive checkpoint files through Control Builder.
- Manual or automatic saves have been completed.

Considerations:

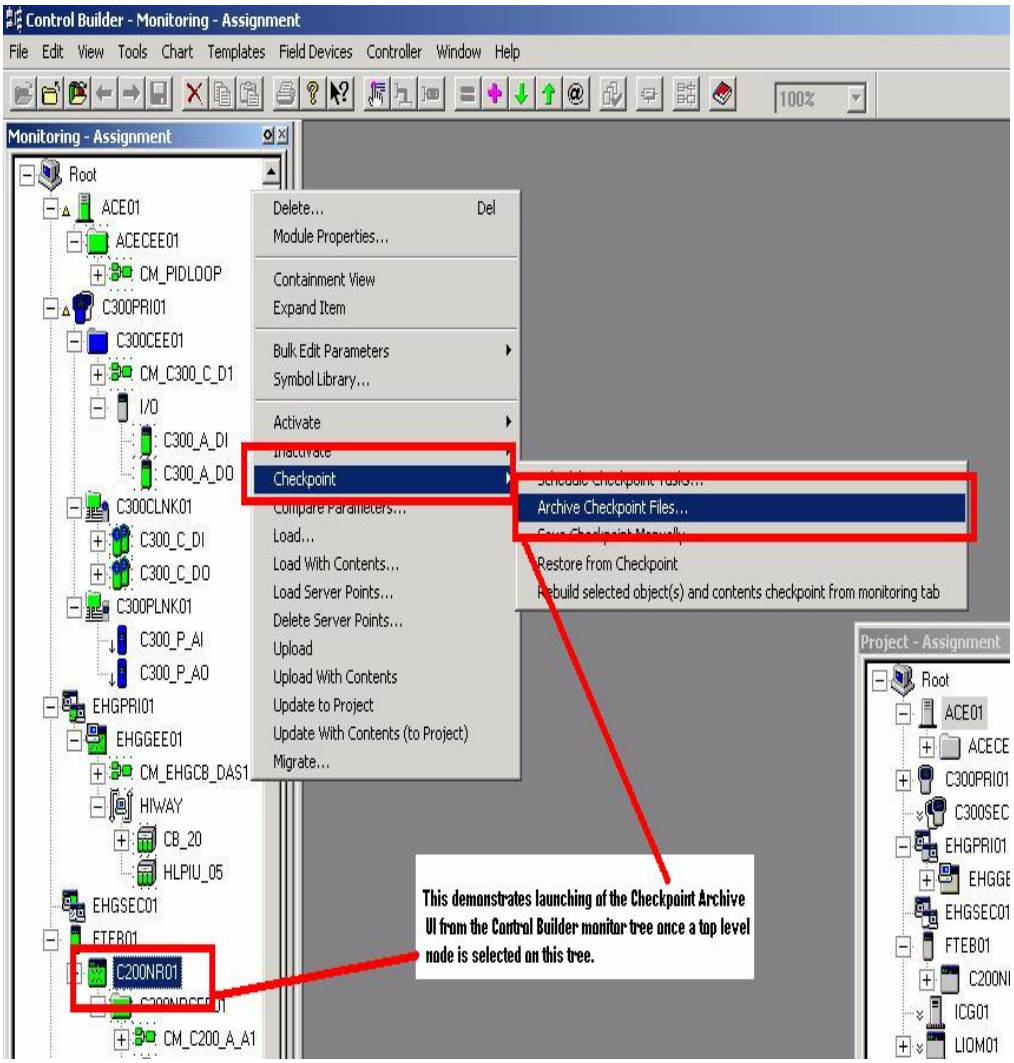
- The Archive operation copies the checkpoint file from source to destination, and leaves the checkpoint file on the source as is.
- You should keep the checkpoint files with the most recent real-time saved information online in the default directories for immediate use. This lets you initiate a checkpoint restore in the minimum amount of time in a disaster recovery scenario. Checkpoint restore has immediate access to the node's latest.cp file for this reason. You may archive the node's latest.cp checkpoint file, but you should never attempt to delete it or remove it from its defaulted directory location.
- The Archive function forces you to define a destination path that is different than the source path.

- You can only set ReadOnly on the checkpoint file copied to the destination. It is never set on the checkpoint file stored on the source
- When specifying the destination path, you can only select the path since you are **NOT** given the ability to change the file name.
- You can archive the selected source file multiple times to the same destination path or to different destination paths, as required.
- If you attempt to archive a source file to a destination path that already contains a copy of the file, follow the existing operating system prompts to overwrite the file.
- If you do choose to overwrite a file in the destination location, you are responsible for determining whether or not the Read Only properties get reset for the file in the destination location.
- If you are using a removable media, be sure the medium has sufficient storage capacity and is ready for a write operation.

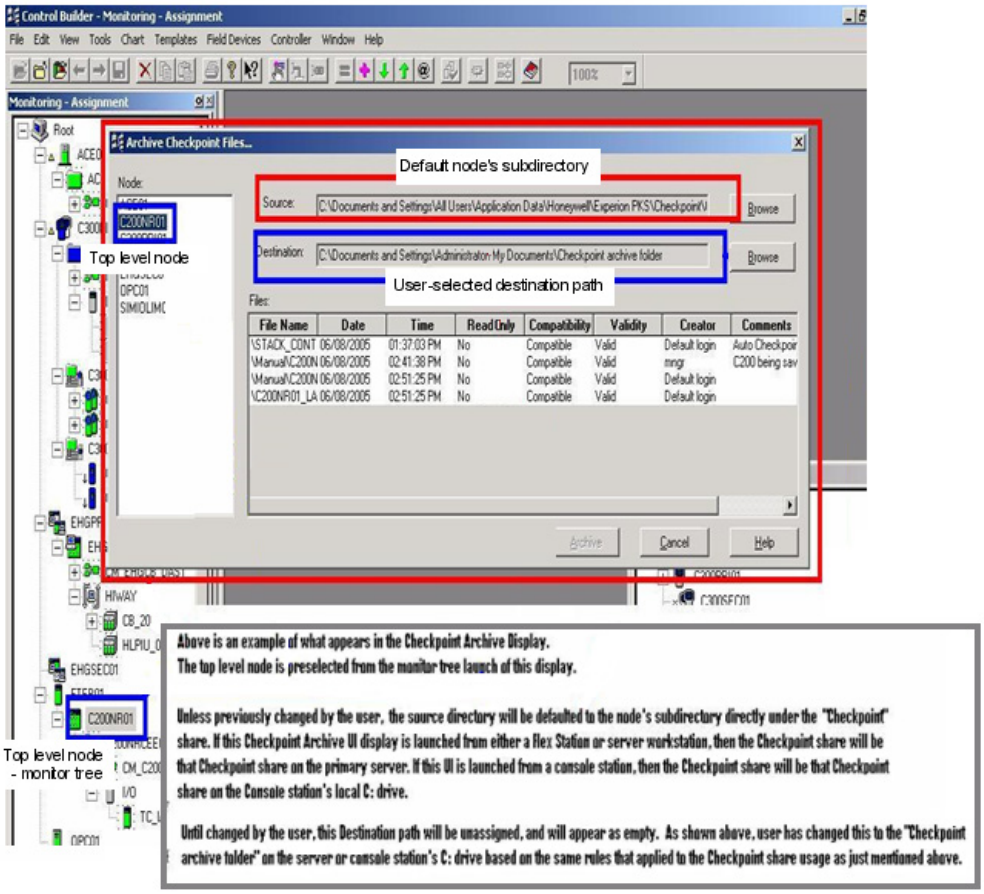
To archive checkpoint files

Step	Action
1	On the **Controller** menu, click **Checkpoint** > **Archive Checkpoint Files**
2	In the **Source** field, be sure the path to the saved checkpoint files is correct. If it is not correct, click the **Browse** button to navigate to the correct location.
3	In the **Destination** field, be sure the path to the archive location for the checkpoint files is correct. If it is not correct, click the **Browse** button to navigate to the correct location.
4	In the **Node** box, select the hardware node whose checkpoint files you want archive. The source path is automatically changed to point to the node's subdirectory in the Checkpoint share. **Tip:** To avoid changing the source path twice when you want to specify a path other than the default one, select the node first and then the source path second.
5	In the **Files** box, select the checkpoint file that you want to archive.
6	• Click the **Archive** button. • Prompt asks if you want to mark the files as ReadOnly. Click the **Yes** button to make files Read Only. Otherwise, click the **No** button.
7	Wait for the copy function to complete.
8	If applicable, repeat Steps 4 to 7 for other nodes.
9	This completes the procedure.

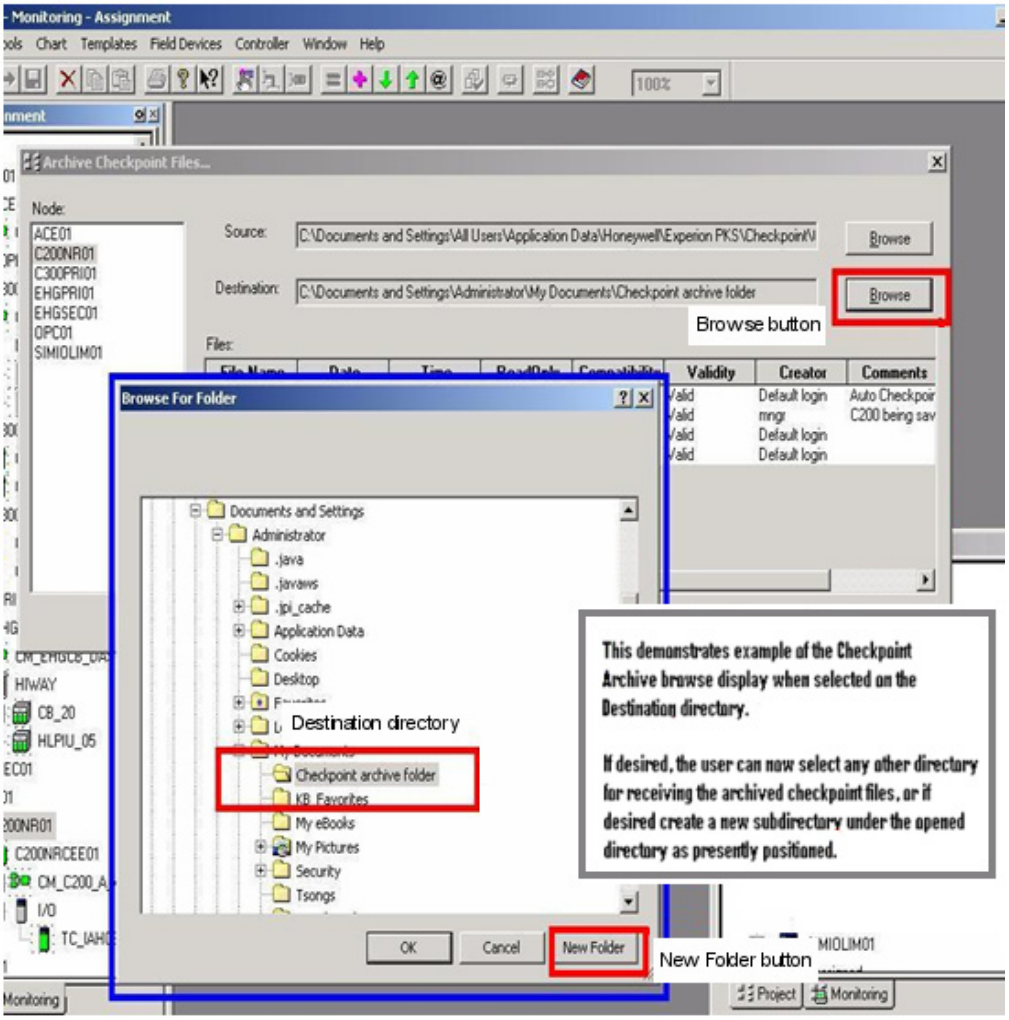
Archive checkpoint graphical reference



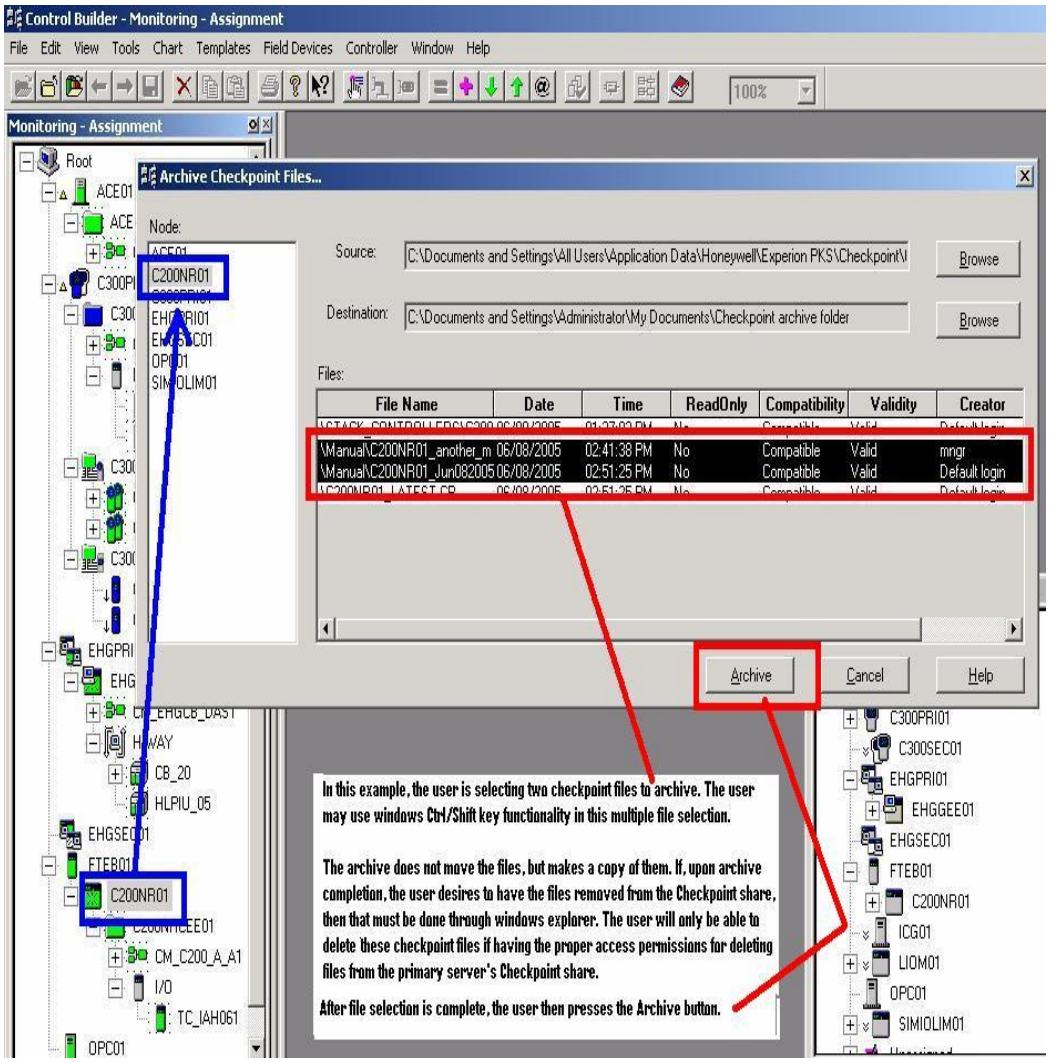
Launching Archive Checkpoint Files by right-clicking node in Monitoring tab and selecting from list



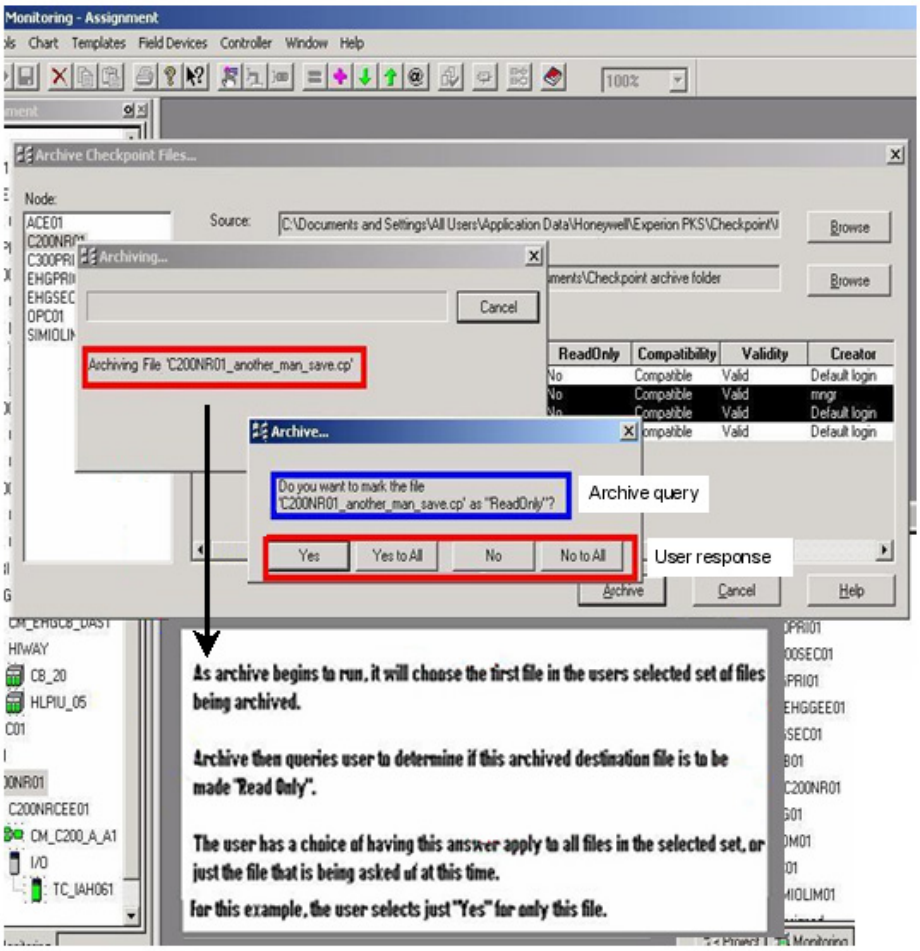
Viewing Checkpoint files for selected node from designated source and archiving to designated destination



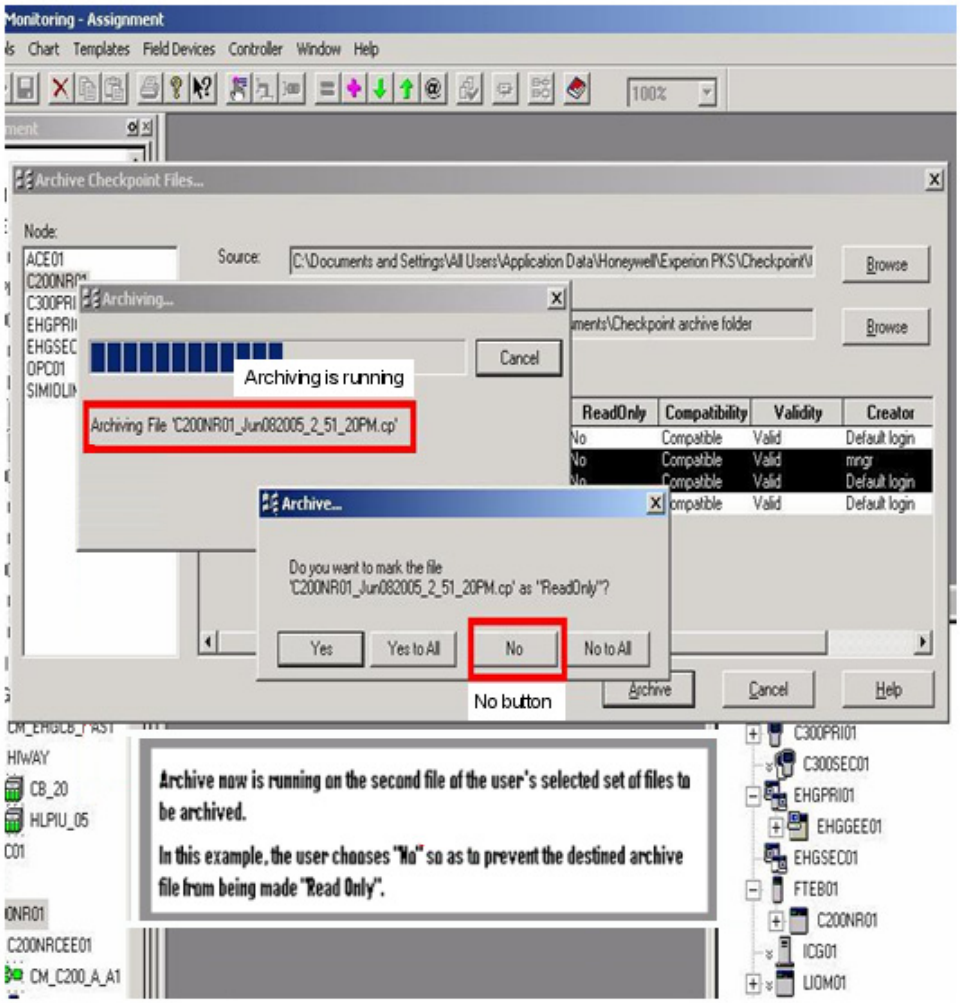
Using browse button to navigate to desired storage location for checkpoint files to be archived



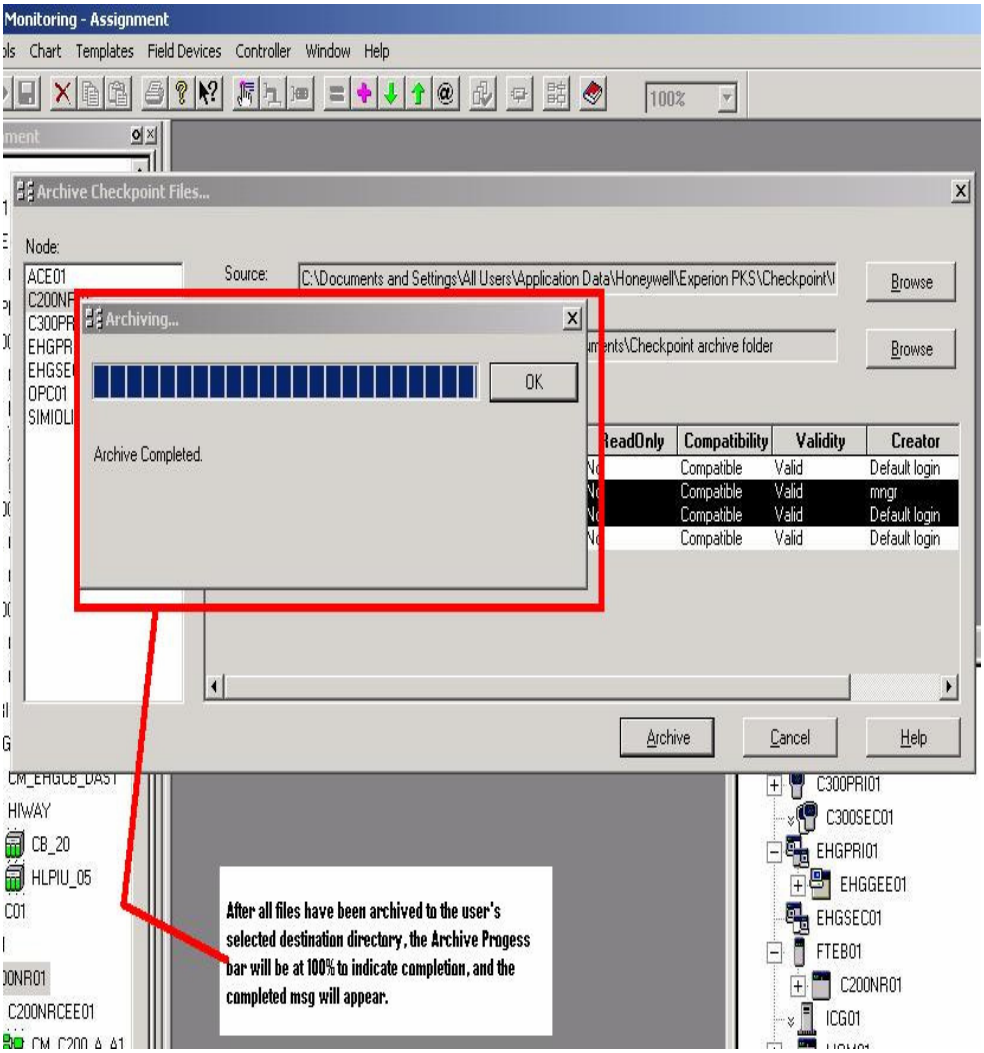
Using Ctrl+Shift keys to select multiple checkpoint files for archiving



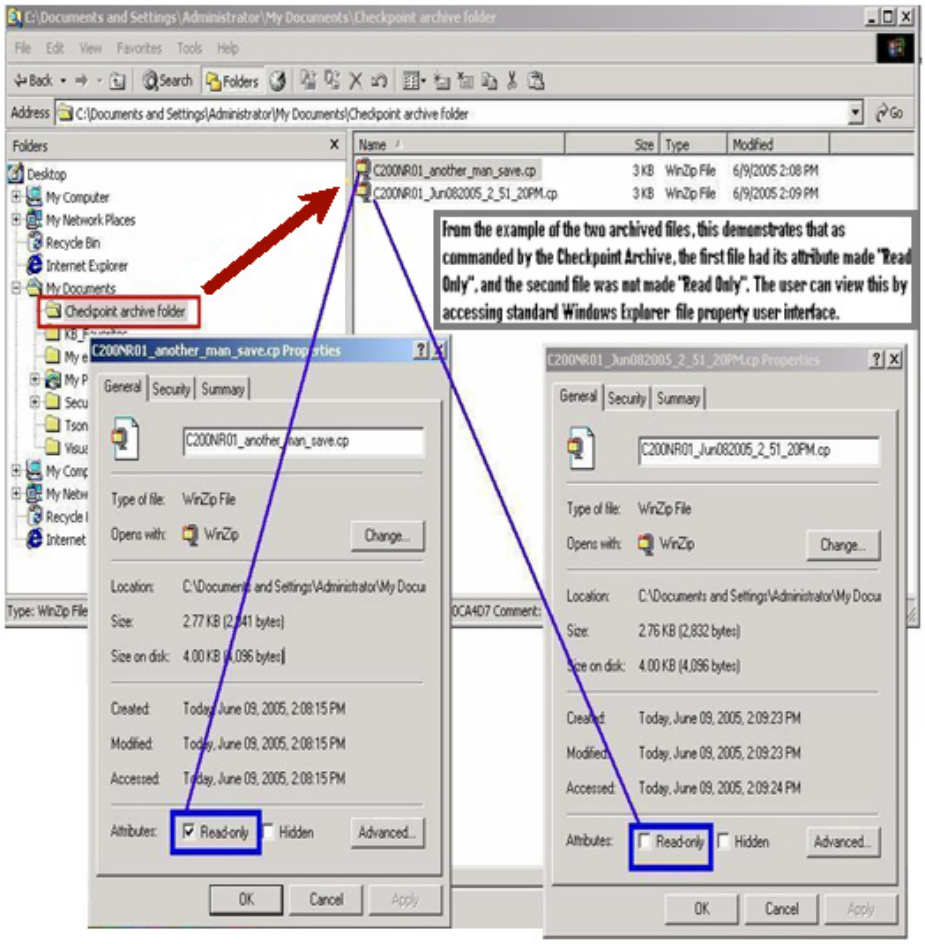
Marking archived file as ReadOnly



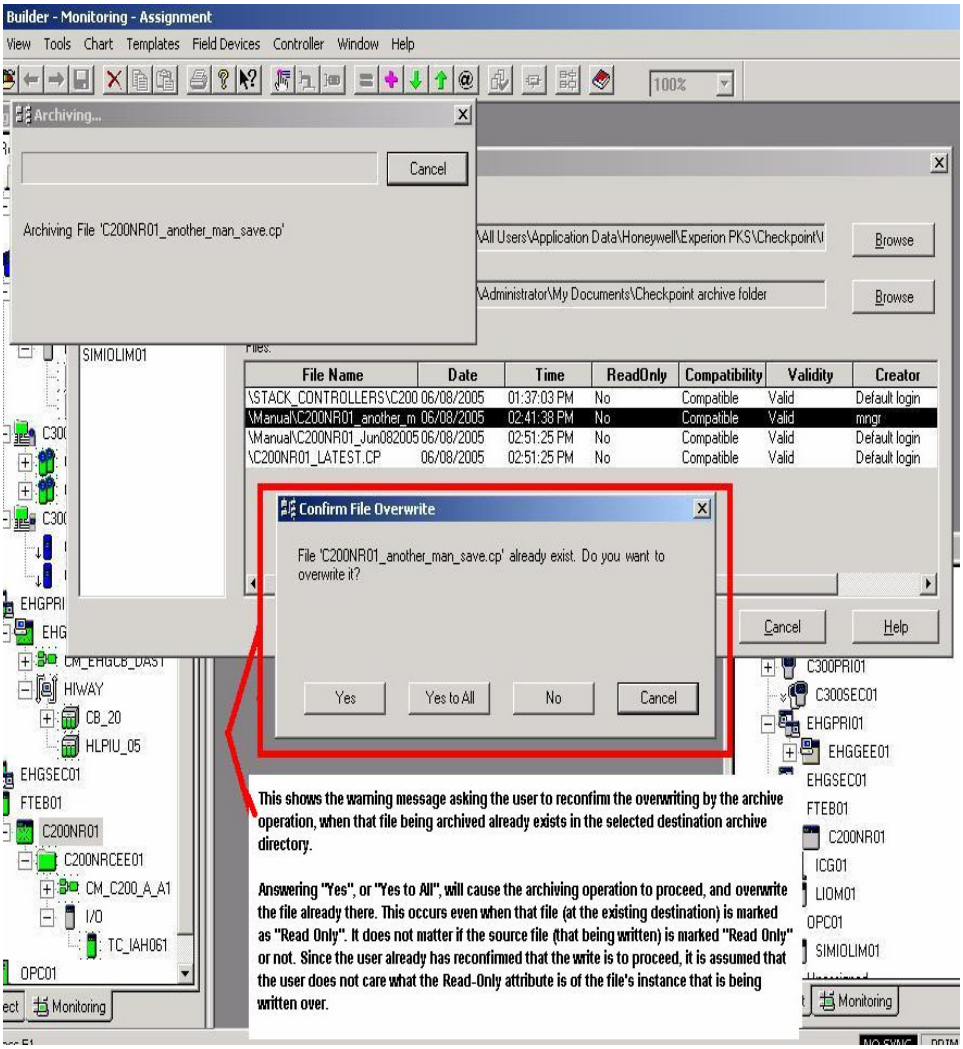
Initiating file archiving without marking file as ReadOnly



Prompt message confirms when file has been archived



Confirming ReadOnly status of archived files



Overwriting an existing archived checkpoint file

Checkpoint file attributes

The Checkpoint functionality defines its own set of file attributes for each checkpoint file. This information is encoded in the checkpoint file and some of this information

appears in the **Files** grid on both the **Restore From Checkpoint** and **Archive** dialogs. The following sections define these checkpoint file attributes.

Compatibility attribute

Definition	Logical integrity of the Checkpoint file data.
States	**Compatible**: The configuration state/instance of all of the blocks and connections among blocks contained in the checkpoint file is consistent with what Experion monitor ERDB and SR knows it to be.
	Incompatible: The configuration state/instance of one or more blocks (as contained in the checkpoint file) is inconsistent with what Experion monitor ERDB and SR knows it to be. The restore of such of file to the controller would most likely result in ghost point operation. A ghost point is defined as a block actually running in the controller, but the user has no visibility to such operation since it is not present in Control Builder's Monitor View. Checkpoint ensures that such a file cannot be used for a checkpoint restore operation.
Notes	1. A node's latest.cp never becomes incompatible. 2. Once a checkpoint file is made incompatible, the action is irreversible. 3. In R300.1, a **Project** load (even without any changes) will cause all prior checkpoint files to be made incompatible. 4. Checkpoint files that are compatible before any **Monitoring** load will retain compatibility after that load

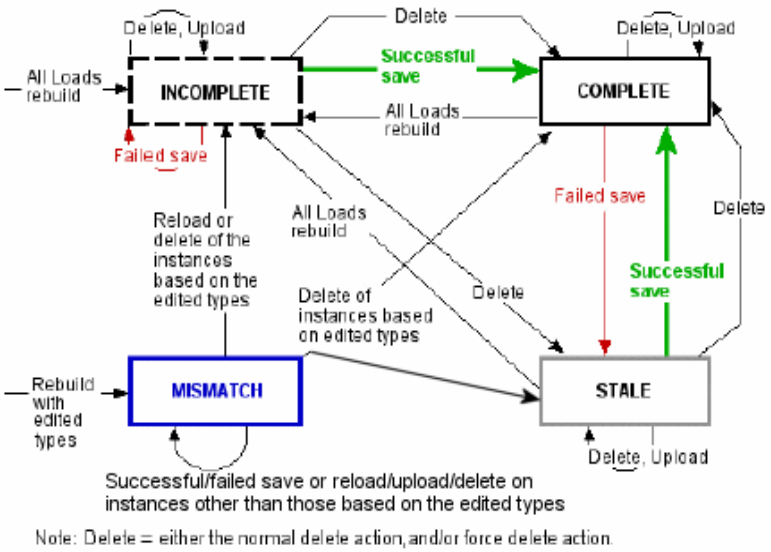
Validity attribute

Definition	Physical integrity of the Checkpoint file data
States	**Valid**: File's computed CRC is equal to the CRC that is stored in the file
	Invalid: File's computed CRC is not equal to the CRC that is stored in the file
Notes	Each checkpoint file has a CRC32 signature which is tested to validate the physical contents before it is used for a restore.

Entirety Attribute

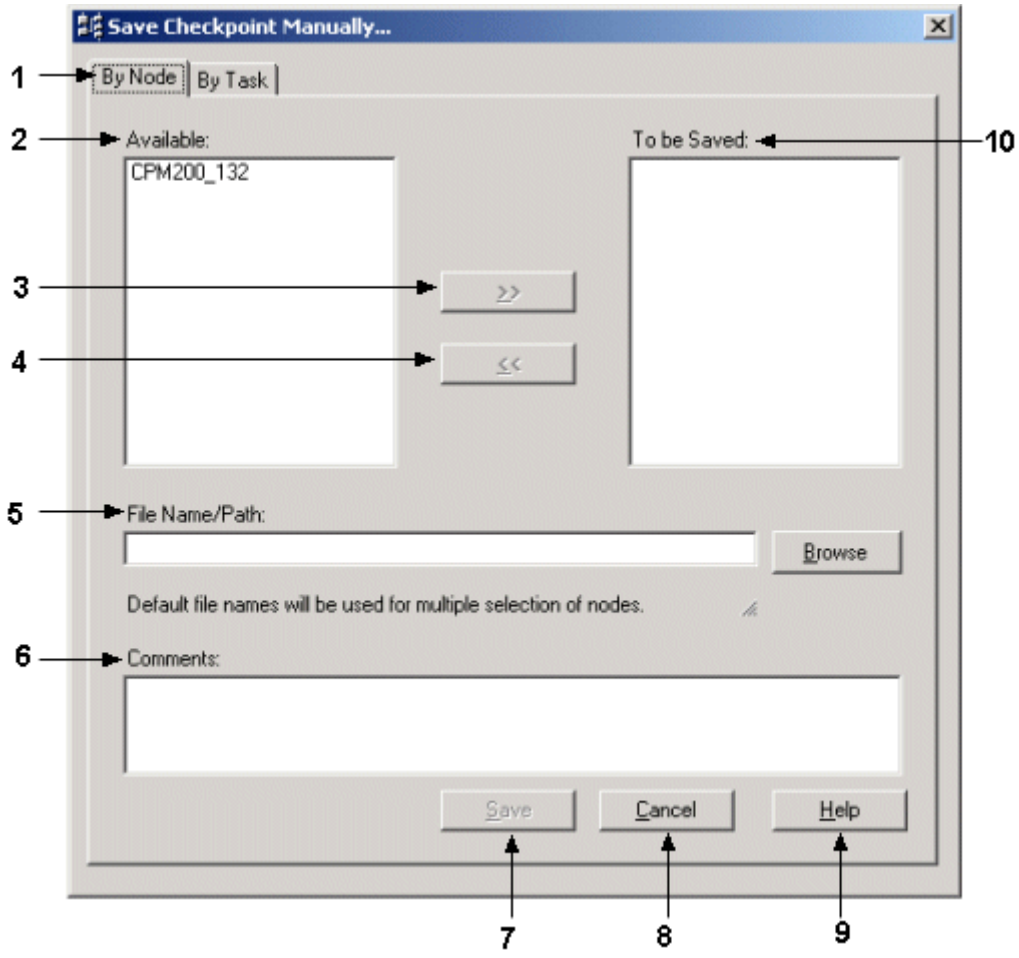
Definition	Completeness of the file with respect to runtime Operational Checkpoint Data (OCD) information.
States	**Complete**: All the tagged objects within the file contain up to date OCD information.

	Incomplete: No OCD information for one or more tagged objects in the file. This could occur after a Load/Reload (even when no changes occurred) of such tagged objects but before a checkpoint SAVE operation. To recover from this condition, you need to perform a successful checkpoint save so that the entirety state becomes complete.
	Stale: One or more tagged objects in the file do not have up to date OCD information. This could occur due to communication, file access or other errors during a checkpoint SAVE operation. In such a case, OCD information is preserved from a most recent previous successful checkpoint SAVE. To recover from this condition, you need to perform a successful checkpoint save so that the entirety state becomes complete.
	Mismatch: This state applies to only Custom type blocks such as CAB, CDB, Phase and any future custom types. For CAB/CDB/Phase: If you change the type, without any subsequent Project/Monitoring loading, and then perform a rebuild checkpoint file from monitor operation, any CMs containing CAB/CDB/Phase blocks (derived from the edited type) will not be consistent with those blocks executing in the controller (based on the type prior to the edit). All such CMs are categorized as mismatch. The only way to make these matched again is a reload from Project or Monitoring tab. Furthermore, for any CM that proceeds to mismatch, all knowledge is lost regarding its former state of complete, stale, or incomplete. However, this does not matter, since you cannot restore a mismatched CM anyway. To recover from this condition, you need to first perform a reload of the mismatched CMs, and then perform a successful checkpoint save so that the entirety state becomes complete.
Notes	1. You can use Checkpoint files with any of the preceding entirety definitions in a checkpoint restore. These files are not corrupted, but are physically fine with regard to CRC checks and compatibility with that information in the System Repository (SR). 2. The Entirety shown for a checkpoint file in the **Restore from Checkpoint** dialog is the overall Entirety of the file. This is based on the highest priority entirety found on all the objects in the checkpoint file. The following is the Entirety priority: – Mismatch (Highest) – Incomplete (Higher) – Stale (High) – Complete (Lowest) 3. In addition to the previously described mismatch state, a load of Custom Type blocks from **Project** or **Monitoring** after a type update causes Control Modules (CMs that include blocks derived from modified CAB/CDB/Phase types) in previous checkpoint files to be not consistent with the Controller. The controller has the newly modified custom block type, but the prior checkpoint files do not.

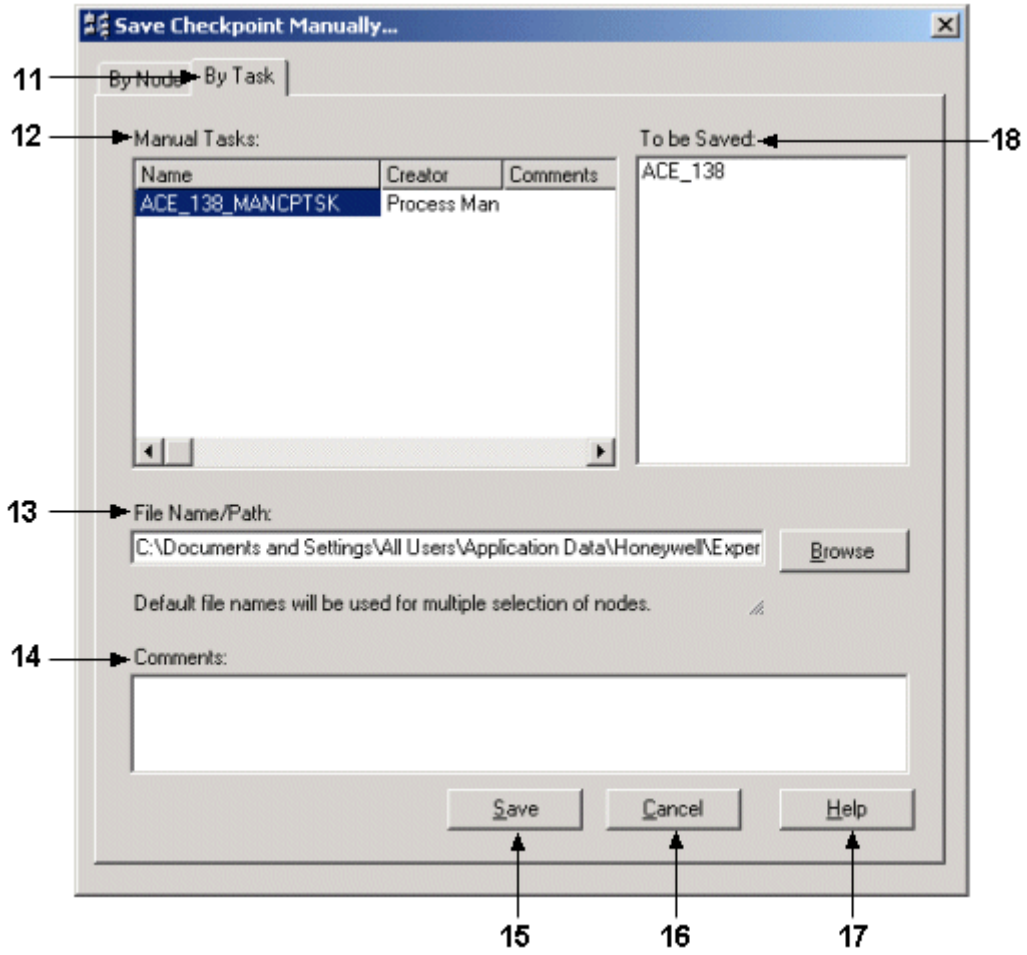
	This is not a mismatch because the previous checkpoint files will be consistent within their set. Unless made incompatible, by a **Project** load, a user may be able to restore such a checkpoint file. In this condition, a **Monitoring** reload will update the custom block type on the monitoring side, but retain compatibility of the older checkpoint files. This permits these older checkpoint files to remain selectable for restore. In this scenario, restoring the former custom block type does not result in any introduction of ghost points, since the same exact block instance gets reloaded and still can be viewed from the monitoring side. The older block instance that is restored may have fewer parameters than that shown in monitoring. Parameters that are shown in the custom block form on monitoring (that were added in the newer block type change that was loaded before the older checkpoint file restore) will show in error on the monitoring form. Although these added parameters will be in error, it still does not result in ghost point operation. 4. When a mismatched file is selected for restore (including node's latest.cp), all of the blocks, with the exception of the mismatched blocks, will be restored to the selected node. The user will need to reload the mismatched blocks from **Monitoring** to have them again reside/operate in the selected node. 5. The following state diagram illustrates how the various actions performed through Control Builder and Checkpoint interfaces change Entirety.
	State transition diagram of node's latest .cp file 

Identifying Functions on the Save Checkpoint Manually Dialog

The following illustrations and callout table identify the functions that are available on the **Save Checkpoint Manually** dialog.

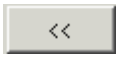


Save Checkpoint Manually by Node Tab



Save Checkpoint Manually by Task Tab

Callout	Function	Description
1	**Node** box	Click tab to move it to the front and use it to initiate a manual Checkpoint by selected node.
2	**Available** box	Shows all currently configured nodes in the **Monitoring** tab. Note that any nodes currently selected in the **Monitoring** tab will also be selected in this box.

Callout	Function	Description
3	 **Add** button	When available, click to add selected available node to **To be Saved** box. Selecting a node in the **Available** box makes the button available. You can use common Window's shortcut keys to make multiple selections.
4	 **Remove** button	When available, click to remove selected node from the **To be Saved** box. Selecting a node in the **To be Saved** box makes the button available. You can use common Window's shortcut keys to make multiple selections
5	**File Name**	Shows the default directory location for the saved checkpoint files. You can enter a desired name of a checkpoint file for a single entity to this path. If you select two or more nodes for checkpointing, this field is grayed out. In this condition, the default name for each node's checkpoint file will be assigned as follows: *<full path to node's checkpoint share subdirectory* \ *Manual* \ *<node name>_<date>_<time>.cp* For both Automatic and Manual checkpoint saves, you can only save Checkpoint Files to the current primary Experion Server's Checkpoint share directory.
6	**Comments**	Type desired comment that applies to all the saved checkpoint files. This field is unavailable when multiple nodes are selected for saving.
7	**Save** button	Click to initiate the save of selected **To Be Saved** nodes to the default or specified checkpoint file. Prompt tells you to view each node's Detail Display Checkpoint Tab to monitor the operation. No progress bar is provided for showing the progression of the checkpoint save operation; since it can take 20 to 30 minutes to complete a checkpoint save for a Controller having a large number of blocks. You can view the status of a given node's checkpoint save by launching that node's detail display. Select the detail display's Checkpoint tab and examine the Checkpoint Save Completion status. The completion can be seen as a journaled event.
8	**Cancel** button	Click to close the dialog and end the current session.

Callout	Function	Description
9	**Help** button	Click to view associated Help information.
10	**To be Saved**	Shows nodes that have been selected for this save operation.
11	**By Task** tab	Click tab to move it to the front and use it to initiate a manual Checkpoint by selected task.
12	**Manual Tasks** box	Show all the scheduled tasks that have been configured through the **Checkpoint Scheduler** with a **Startup Type** of **Manual**. It includes: **Task Name** **Creator** of the task **Comments** configured for tasks, if any.
13	**File Name**	Shows the default directory location for the saved checkpoint files. You can enter a desired name of a checkpoint file for a task with a single node to this path. If you select a task with two or more nodes for checkpointing, this field is unavailable. In this case, the default name for each node's checkpoint file will be assigned as follows: *<full path to node's checkpoint share subdirectory* \ *<manual task name>* \ *<node name>_<date>_<time>.cp* For both Automatic and Manual checkpoint saves, you can only save Checkpoint Files to the current primary Experion Server's Checkpoint share directory. If this is a localized system, you must accept the default name provided by checkpoint, since the file name text box is unavailable and cannot be edited.
14	**Comments**	Type desired comment that applies to all the saved checkpoint files.
15	**Save** button	Click to initiate the save of all nodes associated with the task selected in the **Manual Tasks** box.. Prompt tells you to view each node's Detail Display Checkpoint Tab to monitor the operation.
16	**Cancel** button	Click to close the dialog and end the current session.

Callout	Function	Description
17	**Help** button	Click to view associated Help information.
18	**To be Saved**	Shows nodes that are associated with the task selected for this save operation. You cannot edit this list of nodes here. You must edit the task through the **Checkpoint Scheduler** dialog.

Pre-selection of nodes to be saved

The following table defines how nodes are pre-assigned to the **To be Saved** list box depending on how the save was launched.

If You Launch Manual Save From . . .	Then, Source of Nodes Placed in To be Saved List Upon Display Call up are . . .
Control Builder **Monitoring** tab	The names of checkpoint applicable loaded node(s) selected in the **Monitoring** tab when **Save Checkpoint Manually** dialog is launched appear in the **To be Saved** list. The names of all other checkpoint applicable loaded nodes in the **Monitoring** tab, which were not pre-selected when **Save Checkpoint Manually** was launched, appear in the **Available** list box.
Node's Detail Display on Station	Since the node's detail display is displayed, this implies the node is already pre-selected. The **Save Checkpoint Manually** dialog is launched from the **Checkpoint** tab on the node's detail display. In this case, the node appears in the **To be Saved** list box. The names of all other loaded, checkpoint applicable nodes appear in the **Available** list box. After the launch from the detail display, you can add other nodes from the **Available** list to the **To be Saved** list, as desired, or just initiate checkpoint save of the pre-selected node.

Save arbitration on a given Node

You cannot cancel/abort the manual save process after it has been initiated by clicking the **Save** button. Neither a Save Checkpoint Manually nor Automatic Checkpoint Save operation stops you from doing a load operation to the same node. If a **Project**/**Monitoring** load, or **Monitoring** delete occurs during a **Manual** or an **Automatic** save operation, the saved checkpoint file is marked as **incompatible** immediately after the save has completed. The save operation will also be marked "**failed**". In addition, to keep integrity of the node's latest .cp, this latest.cp is not updated with that information from the **incompatible** checkpoint save. Upon the next **compatible** save, the node's latest.cp is then again updated with the run time information collected by that save.

Execution states for checkpoint save

Checkpoint save ensures the node is in a proper state for saving before it honors the save attempt. Just before any Manual or Automatic save is performed, Checkpoint reads that node's execution state. For most nodes, this is read from the EE block. For node's that do not have an EE, such as OPC and ICG, this is read from the node's platform block.

If the execution state cannot be read or if read and the state is not as expected, then the save does not occur. For a manually initiated save, an error message will be shown and an error will be logged into the Experion error log. For an automatic save, the error will be journaled as a system event in Station.

The following table shows what execution state parameter gets read, and what it is required to be for checkpoint save to proceed.

Node Type	Expected Node Execution States For Checkpoint Save To Proceed	
	Execution State Parameter	***Expected Values***
C200, **C300,** **LIOM,** **ACE,** **SIM-C200,** **SIM-ACE,**	EE block's CPMSTATE	CEEIDLE, CEERUN, CEEIDLESOFTFAIL (when applicable), CEERUNSOFTFAIL (when applicable)
OPC **ICG**	Platform block's OPCGATEWAYSTATE	IDLE, ACTIVE
SIM-IOLIM	Platform block's INITCOMPLETE	COMPLETED

Deferral and Resumption of entity saves during a Manual Save

When the either of the checkpoint manual save operations cause more than 5 nodes to undergo simultaneous saving , the same deferral and resumption of node saves will occur as discussed previously in the *Deferral and Resumption Of Node Saves* section.

Initiating Manual Checkpoint Save

This section provides a procedure for using the Checkpoint User Interface to initiate a Checkpoint save operation manually. You can initiate a manual Checkpoint save either By Node or By Task (Manual).

Prerequisites:

- You have logged on with a security level that permits you to save checkpoint files manually through Control Builder or the Detail Display for the selected node in Station.
- You have loaded control strategies.

Considerations:

- You must have configured scheduled tasks with **Startup Type** of **Manual** through **Checkpoint Scheduler** before you can initiate a **Manual** checkpoint save By Task. **Selection of multiple manual tasks is not permitted.**
- You can initiate checkpoint save operation from a Console Station, Flex Station or Control Builder. A checkpoint file is created in the default directory for each parent node.
- Before you select any hardware nodes for a save, be sure that the given hardware node is in a savable state. See the previous *Execution states for checkpoint save* section for more information.
- Only loaded entities assigned to a selected task are saved.
- Since a Task can have multiple nodes, you can find the Save status for all such nodes on respective node detail displays in Station. In addition, each node will have a separate save completion event in the Event Summary journal.
- If a load or delete operation occurs during a **Manual** checkpoint save operation, the save continues but the file will be marked as incompatible and the save will be marked as failed.

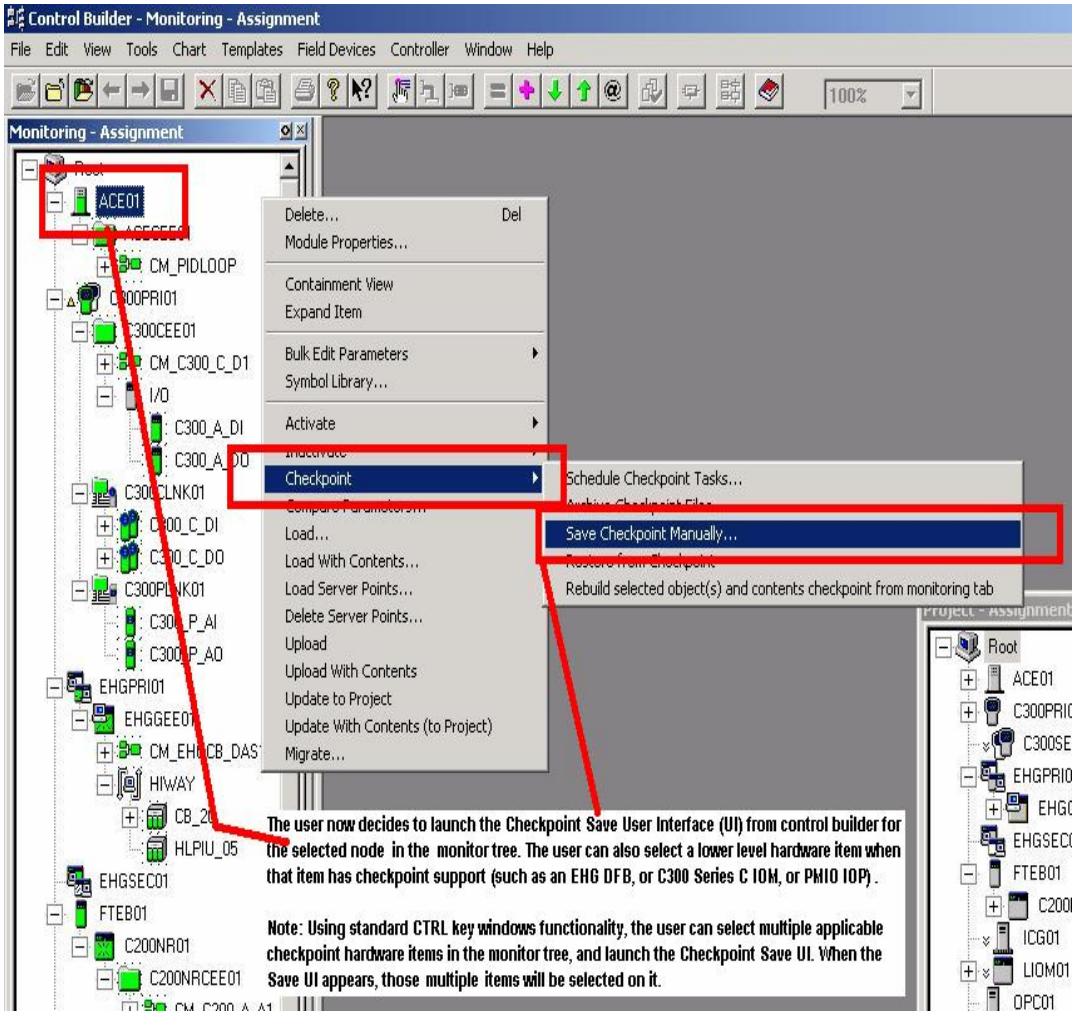
- If a checkpoint rebuild operation occurs during a checkpoint save of a node (does not matter if it is a **Manual** or **Automatic** save), depending on the timing, the save will be marked as failed or may be aborted altogether. The next save that occurs without a simultaneously rebuild/delete or **Project** reload, will again complete successfully apart from any other failures, such as communication errors.
- If there is not enough disk space available to store the saved checkpoint files, the message ***Not able to create checkpoint file, insufficient disk space*** is displayed and an event is generated.
- If you initiate a **Manual** checkpoint save for nodes where an **Automatic** checkpoint save is in progress, the action is accepted and the information from the **Automatic** checkpoint save in progress is used to fulfill the request of the **Manual** checkpoint save.
- If a scheduled task is triggered during a **Manual** checkpoint save operation, the data from the checkpoint files resulting from the just completed manual save will be used to fulfill the saves for each applicable node in the pending **Automatic** checkpoint save task.
- If an **Automatic** checkpoint save is in progress, and a **Manual** checkpoint save and/or another **Automatic** checkpoint save gets initiated for that same hardware node, the checkpoint file resulting from the first **Automatic** checkpoint save will be used to fulfill the save requests for the pending **Manual** and/or other pending **Automatic** checkpoint saves.

To initiate a manual checkpoint save

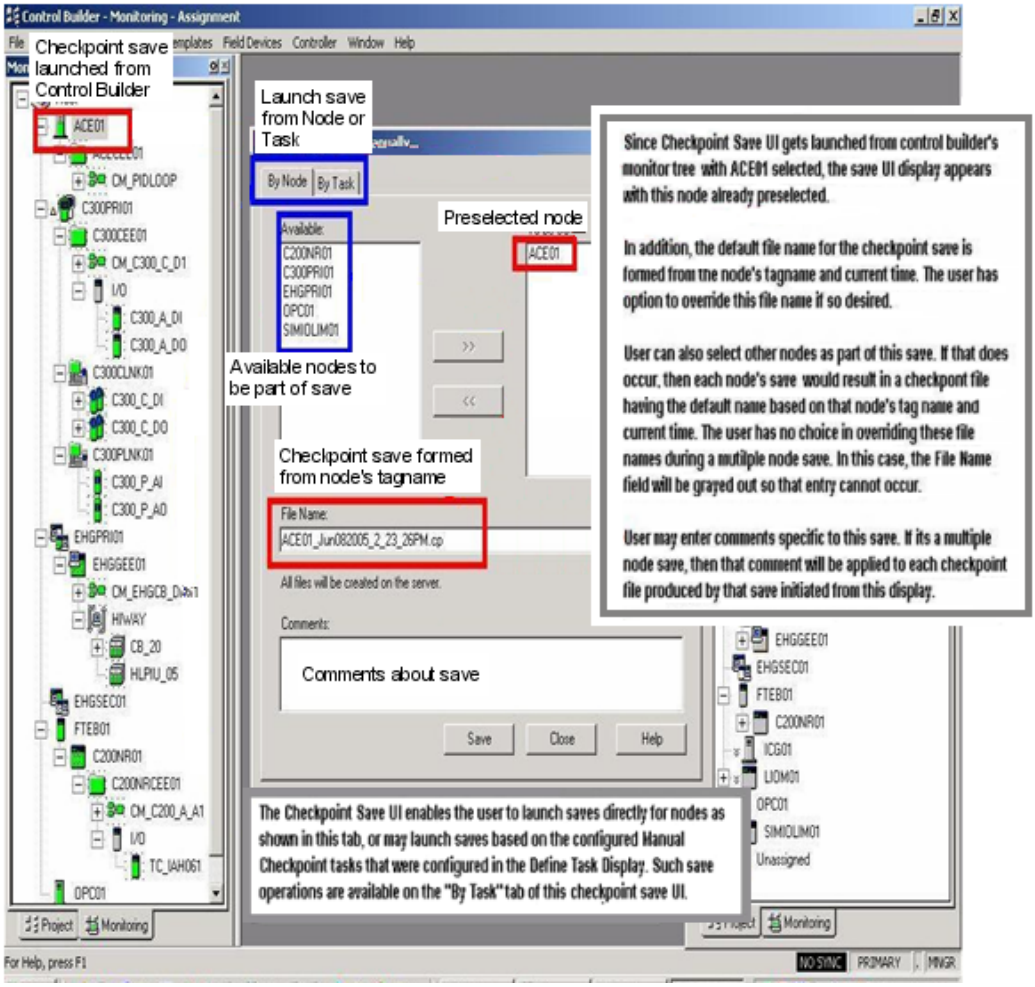
Step	Action
1	On the **Controller** menu, click **Checkpoint** > **Save Checkpoint Manually**.
2	On **By Node** tab, select nodes in **Available** box that you want to checkpoint. You can use common Window's shortcut keys to make multiple selections.
3	Click the  **Add** button to add node selections to the **To be Saved** box.
4	In **File Name** field, be sure directory path and checkpoint file name are correct. Use the **Browse** button to select another directory, if applicable. If you have selected two or more nodes, the default checkpoint file name will be a series of file names for the selected nodes, and this field will be grayed out. For example, if you selected nodes ACE_138 and CPM_132 to be saved, on July 28, 2004 at 8:26:54 PM, the default file names would be:

Step	Action
	C:\Documents and Settings\All Users\Application Data\Honeywell\Experion\CheckPoint\ACE_138/Manual/ACE_138_JUL282004_8_26AM.cp *And* *C:\Documents and Settings\All Users\Application Data\Honeywell\Experion\CheckPoint\CPM_132/Manual/CPM_132_JUL282004_8_26AM.cp*
5	In **Comments** field, type in comment that applies for all checkpoint files, if desired.
6	• Click the **Save** button. • Prompt tells you to see the Save Status display to monitor the operation.
7	Wait for the copy/save operation to complete.
8	If applicable, click the **By Task** tab, select desired scheduled task in the **Manual Tasks** box, and repeat Steps 4 to 7 to initiate checkpoint save for a scheduled task with **Manual Startup Type**.
9	Click the close button to close the dialog
10	This completes the procedure.

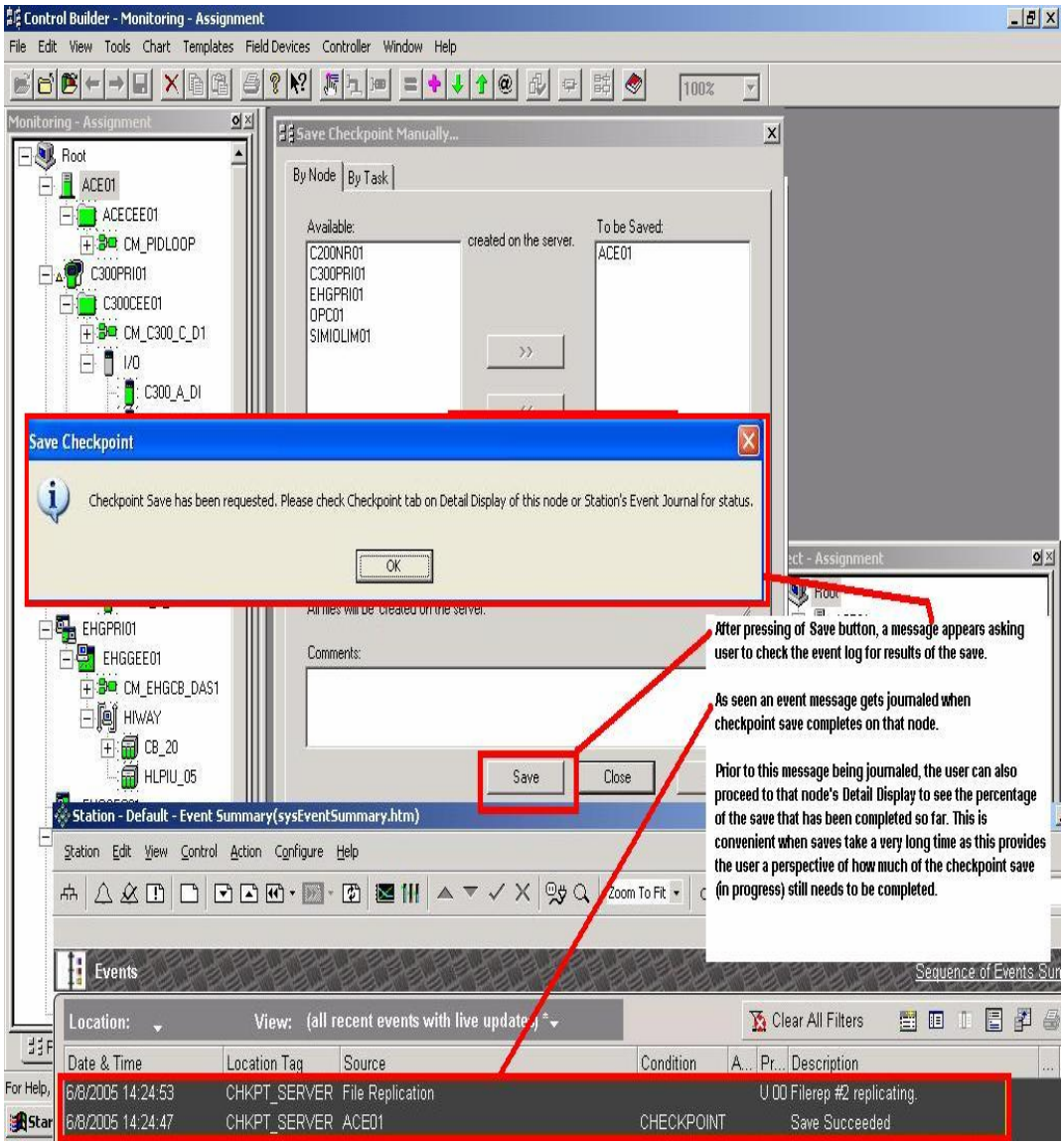
Save checkpoint manually graphical reference



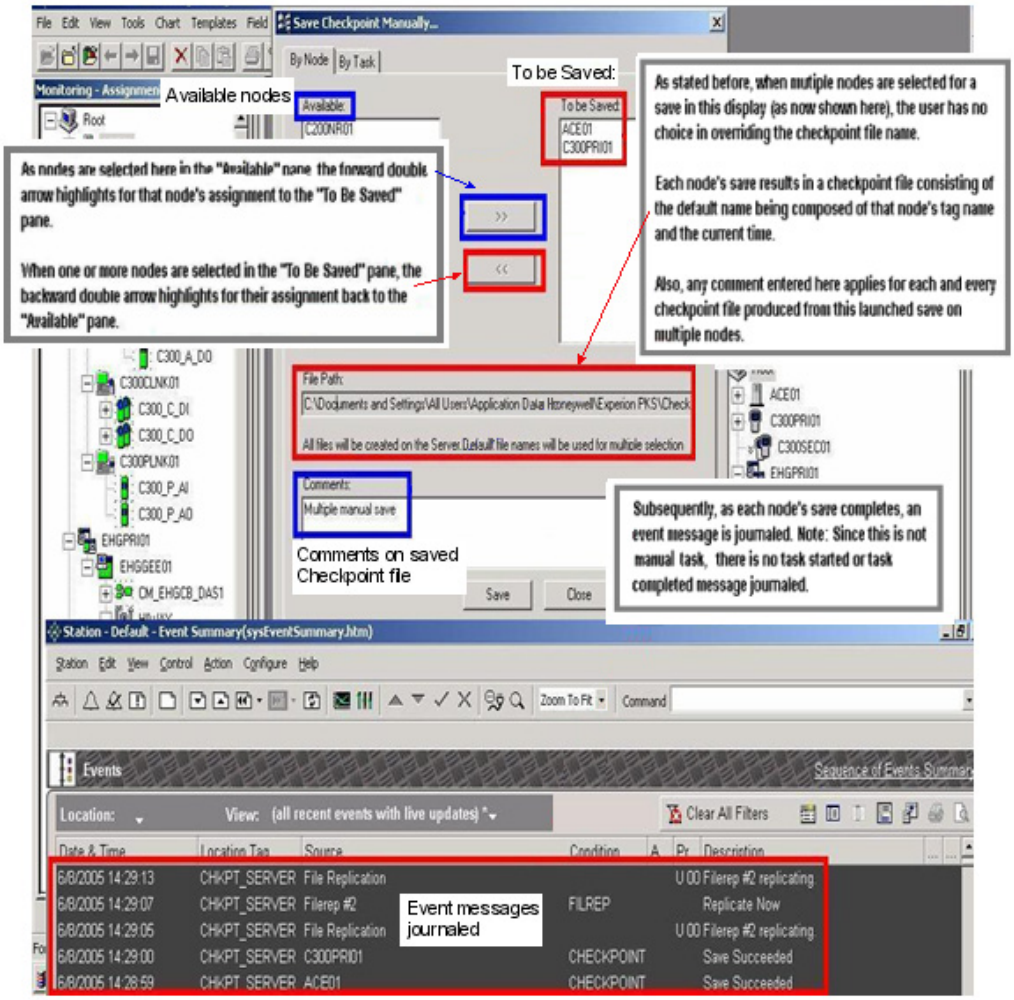
Launching Save Checkpoint Manually by right-clicking selected node and selecting from list - multiple node selections are possible



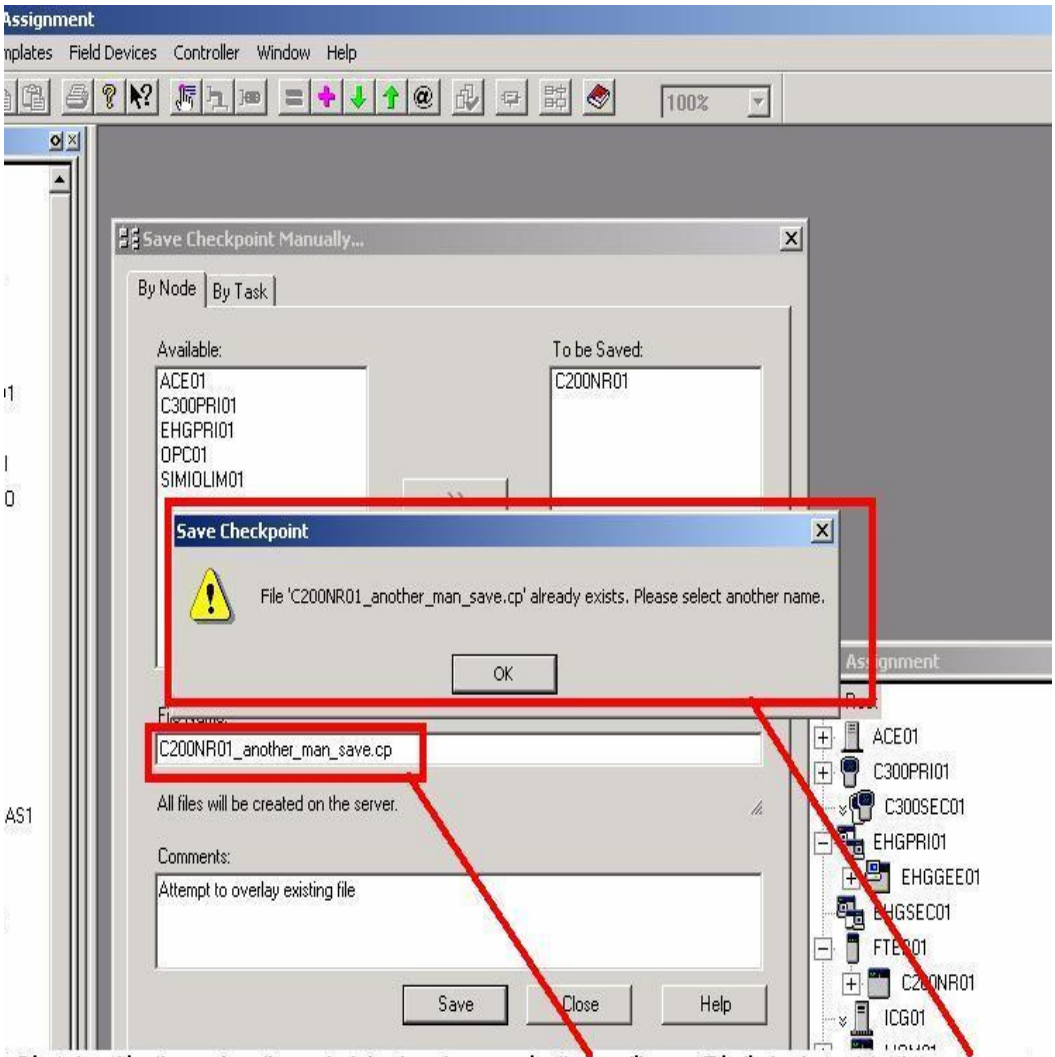
Pre-selected node appears in To be Saved list with default File Name entry and user can enter optional specific comments



Click Save button to manually initiate checkpoint save, click OK button to acknowledge prompt, view status on Node detail display, and check Event Summary for journaled save succeeded event

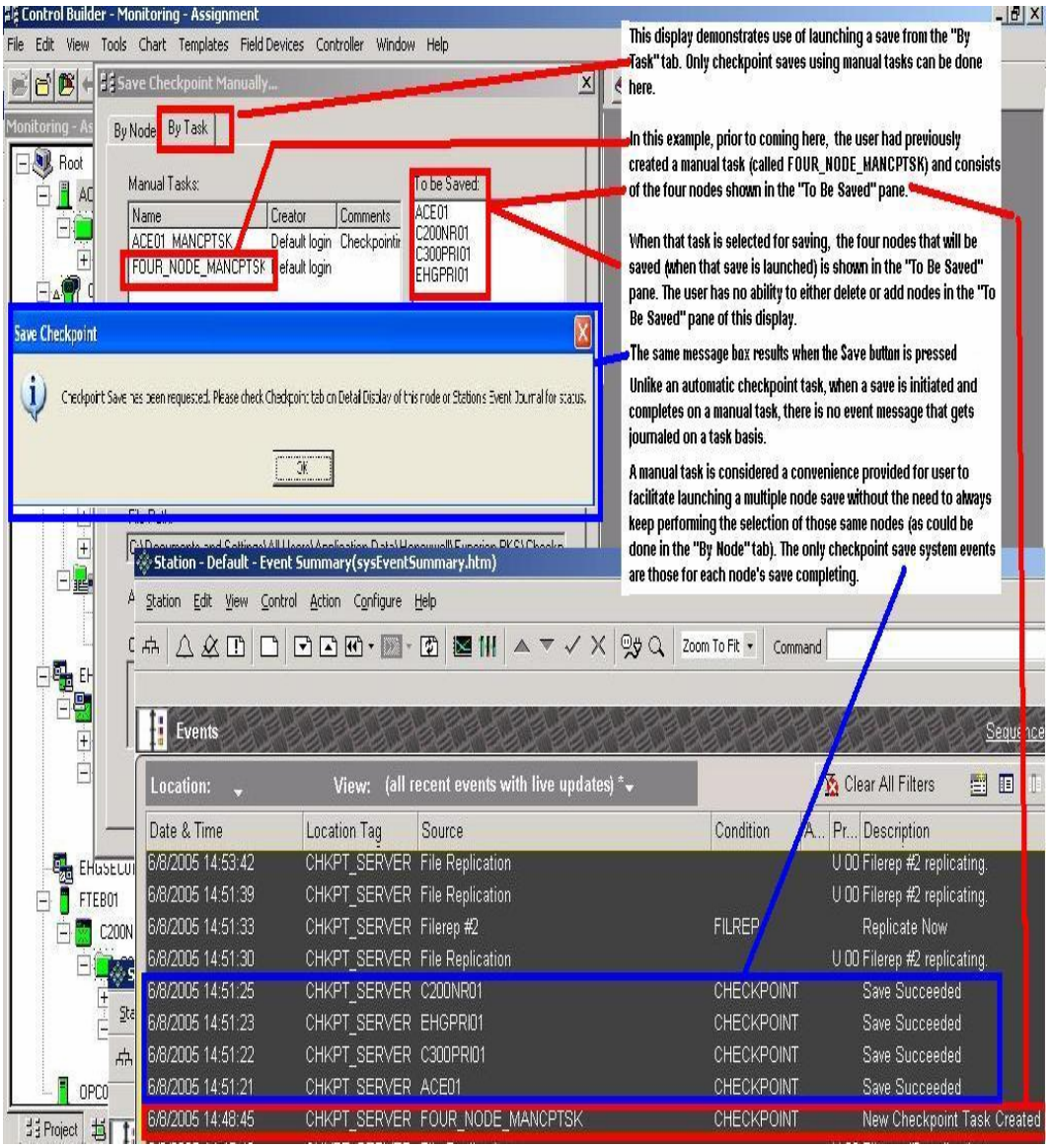


Selecting multiple nodes from Available list results in save actions similar to single node with default path and file names, optional user entered comments, and multiple Save succeeded events journaled in Event Summary

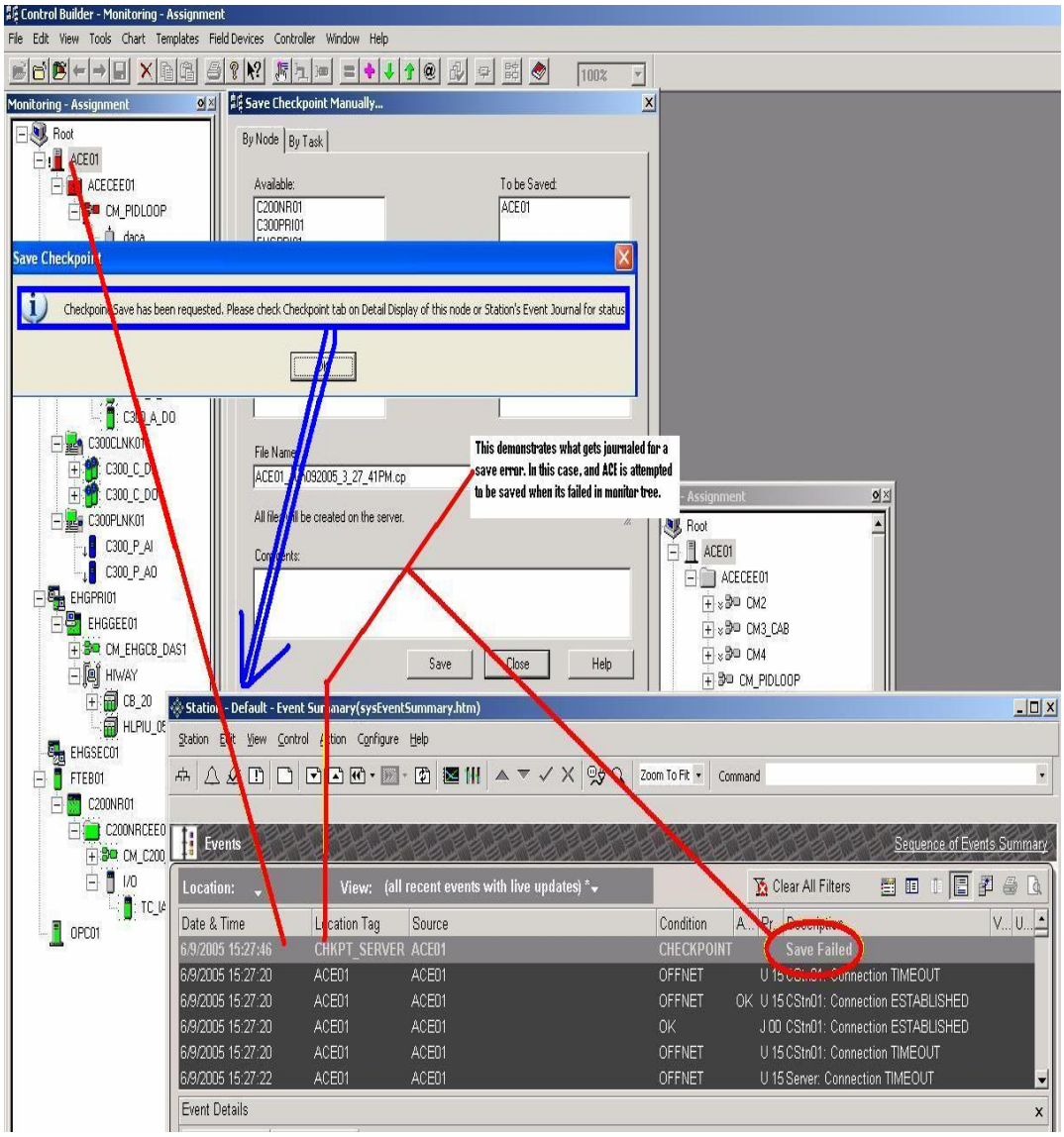


Prior to launching the save here, the user had already made a save using the same file name. This display demonstates the error message given to the user when the checkpoint save is again attempted using that same file name. The user is not given the choice to save into the same file. Another file name must be used. This error message does not get logged to the system event journal.

Prompt warns you if you try to overwrite an existing saved checkpoint file



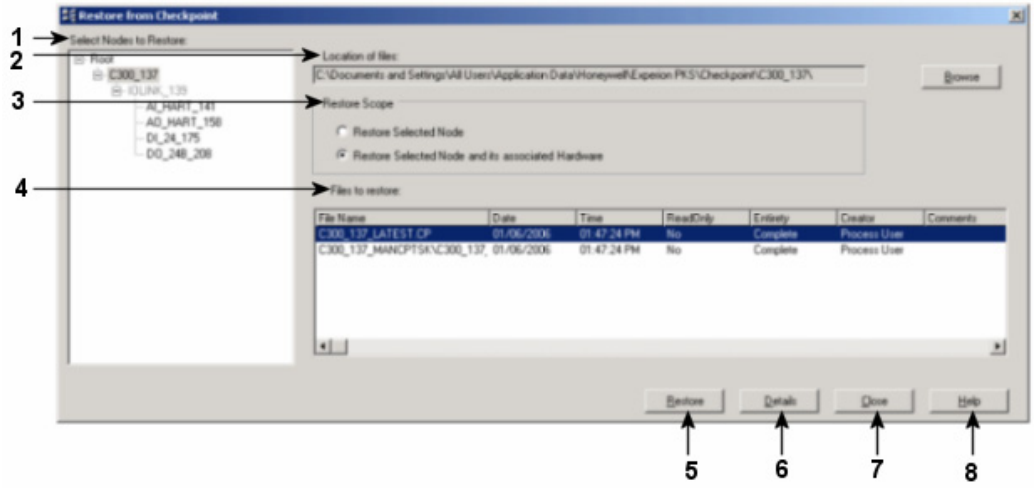
Initiating By Task save results in save actions similar to multiple nodes with default file names and multiple Save succeeded events journaled in Event Summary for each node in the Task - no events get journaled on a Task basis



Example of Save Failed event journaled in Event Summary for manually initiated save

Identifying Functions on the Restore from Checkpoint Dialog

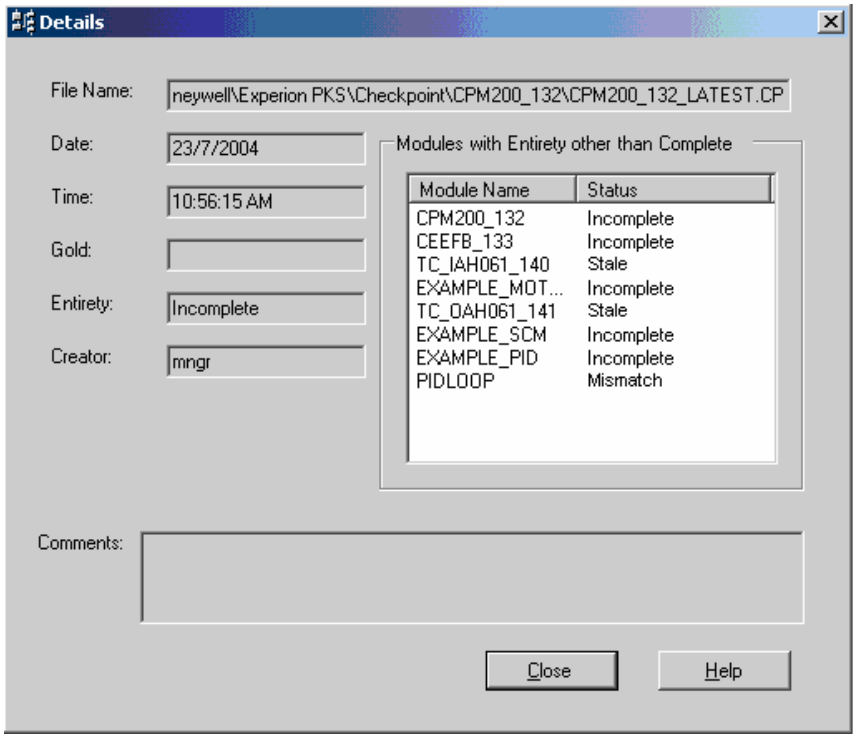
The following illustration and callout table identify the functions that are available on the **Restore from Checkpoint** dialog.



Restore from Checkpoint Dialog

Callout	Function	Description
1	**Select Nodes to Restore** box	Show all nodes in the **Monitoring** tab of Control Builder.
2	**Location of files**	Shows the default checkpoint files directory. If you selected another directory location for the corresponding save operations, use the **Browse** button to select the correct directory location having the checkpoint file(s) desired for restore selection.
3	**Restore Scope** selection	When available, select check box to restore the Selected Node or Selected Node and its associated Hardware.

Callout	Function	Description
4	**Files to restore**	Show all checkpoint files that can support a successful restore operation for the selected nodes. Files that have invalid Compatibility or Validity values are not displayed. A compatible checkpoint file confirms that no structural changes were made to the controller database between the time the controller was loaded from the **Project** tab and the time the controller is restored from the Checkpoint file.
5	**Restore** button	When available, click to initiate the restoration of the selected file. If you select two or more nodes, the files in the **Files for restore** box are unavailable and the latest checkpoint files, such as each node's latest.cp, are used for the restoration. Selecting a checkpoint file or multiple nodes, makes this button available. A Restore progress dialog appears for monitoring the status of the operation.
6	**Details** button	Click to open the **Details** dialog. As shown below, the **Details** dialog shows the modules in the hardware parent node that have incomplete or stale operational checkpoint data (OCD), or mismatch. Refer to the previous *Checkpoint file attributes* section for information about the states for the Entirety attribute.

Callout	Function	Description
		
7	**Cancel** button	Click to close the dialog and end the current session.
8	**Help** button	Click to view associated Help information.

Pre-selection of nodes to restore

The following table defines how nodes are pre-assigned to the **Select Nodes to Restore** list box depending on how the restore was launched.

If You Launch Restore from Checkpoint From . . .	Then, Source of Nodes Placed in Select Nodes to Restore List Upon Dialog Call up are . . .
Control Builder **Monitoring** tab	The names of applicable loaded node(s) selected in the **Monitoring** tab, when **Restore From Checkpoint** dialog is launched, will be highlighted in the **Select Nodes to Restore** list box. The names of all other checkpoint applicable loaded nodes in the **Monitoring** tab, which were not pre-selected at time of **Restore from Checkpoint** launch, also appear in the **Select Nodes to Restore** list, but they are not highlighted. You can launch restore through the selection of multiple lower level hardware children on the **Monitoring** tab, as appropriate to the parent hardware node. When you launch **Restore from Checkpoint** in this case, the lower level hardware items are highlighted in the **Select Nodes to Restore** list.
Node's Detail Display on Station	Since the node's station detail display is displayed, this implies the node is already pre-selected. The **Restore from Checkpoint** dialog is launched from the **Checkpoint** tab on the node's detail display. In this case, the node is highlighted in the **Select Nodes to Restore** list box. The names of all other checkpoint applicable loaded nodes in the **Monitoring** tab, which were not pre-selected at time of **Restore from Checkpoint** launch, also appear in the **Select Nodes to Restore** list, but they are not highlighted. After the launch from the detail display, you can highlight other nodes in the **Select Nodes to Restore** list as desired or just initiate the restore for the pre-selected node. If you want to restore a lower level child item from a detail display, you must launch the restore from the **Checkpoint** tab on that hardware child item's detail display rather than from the node's detail display in Station.

Restore scope selection and actual restore action

The following table summarizes the relationship between the Restore Scope selection on the Restore from Checkpoint dialog and the actual restore action.

Selected Parent Node Type is ACE, C200, LIOM, OPC, or ICG	
Default Restore Scope Selection	**Restore Selected Node and its associated Hardware** (This selection is not user configurable.)
Hardware Item Selection	Only parent node can be selected.
Scope of Restore Action	Restore action always restores full node and all child hardware modules as applicable to that node. This is even the case in SIM-IOLIM although it appears that one could have the option to select child simulated IOPs, but such a restore option is not available.
Selected Parent Node Type is SIM-ACE, SIM-C200, SIM-IOLIM	
Default Restore Scope Selection	**Restore Selected Node and its associated Hardware** (This selection is not user configurable.)
Hardware Item Selection	Only parent node can be selected.
Scope of Restore Action	Restore action always restores full node and all child hardware modules as applicable to that node. This is even the case in SIM-IOLIM although it appears that one could have the option to select child simulated IOPs, but such a restore option is not available.
Selected Parent Node Type is C300	
Default Restore Scope Selection	**Restore Selected Node and its associated Hardware** (Other configurable selection is **Restore Selected Node**.)
Parent Node Hardware Item Selection	Only C300 node is selected. Can select Series C I/O or PM I/O hardware item on associated IOLINK. Refer to the next table topic *Selected Hardware Item Associated with C300 Parent Node* for more information.
Scope of Restore Action	When **Restore Selected Node and its associated Hardware** is selected for C300 node, restore actions occur to entire C300 including all of its Series C I/O and PM I/O IOMs.

Selected Parent Node Type is ACE, C200, LIOM, OPC, or ICG	
	When **Restore Selected Node** is selected for C300 node, Restore action occurs to only C300. This includes just C300 resident data in its C300 CEE, its CMs, SCMs, RCMs, UCMs, its Series A IOMs, and its link EEs. It also includes restoring the needed information residing in the link EEs that expose the Series C I/O and PM I/O IOMs from a C300 perspective.
Selected Hardware Item Associated With C300 Parent Node	
Default Restore Scope Selection	**Restore Selected Node** (This selection is not user configurable.)
Hardware Item Selection	One or more or a mix of Series C I/O and PM I/O IOMs selected under the IIOLINK(s) of a C300 parent.
Scope of Restore Action	Restore action occurs to the selected Series C I/O and PM I/O IOM children. This includes restoring their needed information residing in the C300 link EEs which exposes the IOMs from a C300 perspective.
Multiple Node Platforms Selected - For Example, ACE, C200, and C300 All Selected	
Default Restore Scope Selection	**Restore Selected Node and its associated Hardware** (This selection is not user configurable.)
Hardware Item Selection	More than one parent node is selected. In this condition, no children of any parent node can be selected.
Scope of Restore Action	Restore action occurs to every selected parent node using that node's latest.cp. The restore action always restores the full parent node (including all child hardware modules as applicable to that parent node).

Restoration file display reference

The following table defines what files are displayed in the **Files to restore** list box depending on whether one or more nodes are selected, or one or more child hardware modules are selected in the **Select Nodes to Restore** list.

If Selection in Select Nodes to Restore List is . . .	Then, Files to restore List Displays . . .
Of only one parent node.	Has all files corresponding to the selected parent node only.
Of more than one parent node.	Is unavailable. Shows only the **LATEST** file message. Each given node's latest.cp file is implicitly selected for restore for each selected node
Of the hardware children located under only one node. The selected hardware children can be mixed from among the various link EE sub-parents under the given node.	Has all files corresponding to the children's parent node only.
Of the hardware children located under multiple nodes.	Is unavailable. Shows only the **LATEST** file message. The latest.cp of each applicable parent node (having selected hardware children) is implicitly selected for restore information for those for selected hardware children.

Order of restore checks

Checkpoint restore automatically performs certain tests on a node and the checkpoint file selected before the restore attempt is honored. Checkpoint then asks you to manually confirm/approve of certain conditions. Once you approve the conditions, the restore proceeds on the selected node. The following table summarizes the order of these checks and the needed confirmations/approvals.

Checkpoint Test or Confirmation	Description of Checkpoint's Action and/or User Action
First test - IDLE Check	Prior to selecting any hardware nodes for restore, you must confirm that the given hardware node is in the correct state. The Checkpoint Restore function will automatically test for the permitted restore states as noted in the following *Execution states for checkpoint restore* section. If node is not in one of the permitted states, the Checkpoint

Checkpoint Test or Confirmation	Description of Checkpoint's Action and/or User Action
	users interface rejects the restore attempt and generates an error message as follows.  An additional state is permitted for checkpoint restore to be initiated on any Windows based platforms, such as ACE, OPC/ICG, and any simulation environment. This state is when the node appears failed (in Red) on the **Monitoring** tab. In this condition, a restore attempt is honored since the platform block must be sent for the CDASP server to start the platform's Windows based executable (.exe). After this occurs, the checkpoint restore will wait until the running .exe is ready to accept the remaining part of the checkpoint restore. When that is sensed, the checkpoint restore then proceeds to completion. If a non-redundant embedded controller, such as C200, C300, or LIOM icon is in red on the **Monitoring** tab, the user must first recycle power to the controller so that it restarts in a NOTLOADED (no database) (yellow icon) state before proceeding with the checkpoint restore. If the restore is targeted for just the selected hardware child items of that node, then the preceding state tests revert to just ensuring the node is in an IDLE or RUN condition, as noted in the previous *Execution states for checkpoint save* section. The child hardware module must be in required restore states as noted in the following *Child hardware restore checks* section. The Checkpoint Restore user interface will let the restore proceed, if the child item is not in the required state. The module itself will reject the restore attempt. In this case, error messages are posted in the Restore Progress bar dialog to alert users.
Second Test – Checkpoint File Entirety Check	If the checkpoint file selected for restore does not have an Entirety state of complete, the user is warned that the file possibly has inconsistent run time information in it. If the Checkpoint Restore detects that the selected checkpoint file does not have **Complete Entirety**, it will display the following message requesting the user to reconfirm if the restore attempt is to proceed:

Checkpoint Test or Confirmation	Description of Checkpoint's Action and/or User Action
	 Although such a file is logically compatible with the configuration/connection information in the Monitoring tab and System Repository, be sure you carefully consider the Entirety Attribute states in the previous Checkpoint file attributes section before proceeding with such a restore. If you have not previously viewed the Checkpoint Restore Details pane for this file selection, click the **Cancel** button to go back and call up this selection's Details pane. This lets you see which tagged blocks are not in a ***Complete*** entirety state and will be restored in such state(s).
Third Test – Ensure there is no other operation that is modifying the node's database.	Checkpoint asks the user to confirm that no other operation is running that can modify the database of the node selected for restoration through the following message.  Users can suppress this warning message by selecting the **Do not show this message** again check box before clicking the **Continue** button.
ATTENTION	

Checkpoint Test or Confirmation	Description of Checkpoint's Action and/or User Action
	You must initiate the restore by clicking the **Restore** button for the controller release validation tests against the selected checkpoint file to occur. If the release checks fail, the controller is left undisturbed. However, the user would need to select another file for the restore.
Fourth Test – Extended release/validity tests	Ensures that the checkpoint file selected for the restore is compatible with the node to be restored. This test goes beyond just checking for structural (logical) compatibility and physical file integrity. The additional validation tests include the following. • Ensure checkpoint file matches the controller firmware/program release version. Checkpoint reads the release from the controller and compares it to the one in the checkpoint file. For a Window's platform where the node is in a failed state (red icon in **Monitoring**), Checkpoint attempts to restore just the platform block to get the platform's .exe running. Once the .exe is running, Checkpoint restore queries for the program's release version of the .exe that is now executing. • Ensure checkpoint file matches the current Experion release installed. Checkpoint checks that the Experion release version saved in checkpoint file matches the Experion release version of the checkpoint logic.dll deliverable. This deliverable is always needed in the restore process. • Ensure checkpoint file's format version number matches the one expected by the Experion release. If the version does not match, the Checkpoint restore code does not know how to parse the checkpoint file structure. • Global GUID test – This ensures that the checkpoint file selected for restore is one that was actually saved from this node. This ensures that a user **cannot** use this checkpoint file to restore an identically named node in some other server cluster. Once the preceding tests are verified, the checkpoint restore proceeds to send the restore information to the node. If the checks fail, an error message appears in the Restore Progress dialog and the same error is logged to the Experion error logs.

Execution states for checkpoint restore

The following table shows what execution state parameter gets read, and what it is required to be for checkpoint restore to proceed.

Node Type	Expected Node Execution States For Checkpoint Save To Proceed	
	Execution State Parameter	***Expected Values***
C200, C300, LIOM, ACE, SIM-C200, SIM-ACE,	EE block's CPMSTATE	NOTLOADED (this is No Data Base), NOCEE, CEEIDLE CEEIDLESOFTFAIL (when applicable),
OPC ICG	Platform block's OPCGATEWAYSTATE	IDLE, OPC Error
SIM-IOLIM	Platform block's INITCOMPLETE	NOTCOMPLETED COMPLETED

Child hardware restore checks

The following table shows how the checkpoint restore applies when a node's child hardware modules are either individually selected for restore or when they are restored as part of the full node's restore.

Series C or Process Manager I/O Modules on C300 I/O Link		
Restore Scope And Selected Hardware Item(s)	***Child Icon State on Monitoring Tab***	***Restore Behavior***
Restore Scope = "Restore Selected Node and its associated Hardware" Only C300 is selected in "Nodes To Restore Box" No child hardware items are selected.	Blue	Idle - Child module restores with no errors or warnings.
	Green	Run - Full restore skips over child, warning is posted that child was not restored.
	Yellow	NODB - Child module restores with no errors or warnings.
	Red	Failed – May or may not succeed. See **Note 1**.
	Gray	CDA not operable – restore attempt will not succeed.
Restore Scope = "Restore Selected	Blue	Idle - Child module restores with no errors or warnings.

Series C or Process Manager I/O Modules on C300 I/O Link		
Node" **One or more child modules are selected in "Nodes To Restore Box"**	Green	Run - Restore skips child, error is posted that child was not restored.
	Yellow	NODB - Child module restores with no errors or warnings.
	Red	Failed – Not permitted. See **Note 2.**
	Gray	CDA not operable – restore attempt will not succeed.

Notes:

1. The module is either powered OFF or the link EE does not have the module's configuration information. If the module is powered off, the user must power it back on. This will cause the module to proceed to its NODB (yellow) state. If the module is already powered, the user can just initiate the full C300 restore. This will first re-establish the C300's link with the information that will expose the module to the CDA server. Then, the restore automatically proceeds on the module itself.

2: The module is either powered OFF or the link EE does not have the module's configuration information. If the module is powered off, the user must power it back on. This will cause the module to proceed to its NODB (yellow) state. If the module is already powered, the user must either reload the module or initiate a C300 only restore to get the given link EE configured with the information that will expose the module to the CDA server.

Checkpoint restore/save arbitration with other database changing operations

Checkpoint save or restore operation can occur during any one of the following operations on a node.

- Load from Project or Monitoring tab,
- Delete from Project or Monitoring tab,
- On-Process Migration (OPM) operation,
- Upload to Monitoring,
- Initiate another checkpoint restore or save operation, or
- Another checkpoint restore or save operation is in progress.

If a saved checkpoint file results from any of the previous operations, checkpoint must ensure the integrity of the saved file. If a checkpoint restore occurs on a node while an upload or save is occurring, the final restored state of the node must have the identical CCD/OCD information that was present in the node when the checkpoint save was done.

The following table shows how each of the operations is arbitrated correctly.

Operation In Progress	**Action Taken for Given Initiated Operation**					
	Checkpoint Save	Checkpoint Restore	Rebuild Checkpoint or Upload	Load (Project/ Monitoring)	Delete from Monitoring	Controller OPM
Checkpoint Save	Checkpoint Service arbitrates the in progress and pending saves. (See Note 1-1)	In progress checkpoint save becomes either stale or aborts. (See Note 1-2)	Rebuild and Upload has no errors. In progress checkpoint save completes but may or may not have errors (See Notes 1-3 and 1-5)	In progress checkpoint save becomes incompatible (See Notes 1-4 and 1-5)		
Checkpoint Restore	In progress checkpoint restore completes, save becomes stale or aborts (See Note 2-1)	User must have site practice to arbitrate this. If not arbitrated by user, then node can end up with a corrupt database.	Both restore and rebuild complete without error. However, upload will most likely fail since blocks it may be accessing may not exist yet in node. Furthermore, monitor ERDB could become corrupted. User must have site practice to arbitrate this. If not arbitrated by user, then monitor can end up with a corrupt database.	Load, delete, and OPM will arbitrate among themselves as they use an ERDB lock for this. However, user must have site practice to arbitrate any of these operations with an ongoing restore. If not arbitrated by user, then node can end up with a corrupt database.		
Rebuild Checkpoint and Upload from Controller	Rebuild/upload operation completes, initiated checkpoint save aborts (See Note 3-1)	Both rebuild/upload and restore completes without error	The request for another operation to be initiated here can only occur from another Control Builder. Arbitration already handled in Experion by ERDB locks and/or other mechanisms. These operations will not be honored until the Rebuild that is already in progress completes.			
Load from	Load	User must	The request for another operation to be initiated can only occur from			

Operation In Progress	Action Taken for Given Initiated Operation					
	Checkpoint Save	Checkpoint Restore	Rebuild Checkpoint or Upload	Load (Project/ Monitoring)	Delete from Monitoring	Controller OPM
Project or Monitoring	operation completes, initiated checkpoint save aborts (See Note 3-1)	have site practice to arbitrate this. If not arbitrated by user, then node can end up with a corrupt database	another Control Builder. Arbitration is already handled in Experion by ERDB locks and/or other mechanisms. These operations will not be honored until the Load that is already in progress finishes.			
Delete from Monitoring	Delete operation completes, initiated checkpoint save aborts (See Note 3-1)	User must have site practice to arbitrate this. If not arbitrated by user, then node can end up with a corrupt database.	The request for another operation to be initiated here can only occur from another control builder. Arbitration already handled in Experion by ERDB locks and/or other mechanisms. These operations will not be honored until the delete from Monitoring that is already in progress completes.			
OPM	OPM operation completes, initiated checkpoint save aborts (See Note 3-1)	User must have site practice to arbitrate this. If not arbitrated by user, then node can end up with a corrupt database	The request for another operation to be initiated here can only occur from another control builder. Arbitration already handled in Experion by ERDB locks and/or other mechanisms. These operations will not be honored until the OPM that is already in progress completes.			

Notes:

1-1 The following summarizes what happens when multiple save requests are pending for a given node:

- If an auto-checkpoint save is in progress and the user initiates a manual save for nodes where the auto save is in progress, it is accepted and information from such an auto save in progress is used for the manual save after the auto save completes.
- If an auto-checkpoint save is in progress, and another task scheduled for the same node gets triggered, the results of the auto-save just finishing will be used to complete the pending request. This approach prevents unnecessary accesses to the EE.
- If a manual save is in progress, and a scheduled task is triggered, the resulting checkpoint data from the manual save just performed will be used to complete the pending auto-checkpoint

Operation In Progress	Action Taken for Given Initiated Operation					
	Checkpoint Save	Checkpoint Restore	Rebuild Checkpoint or Upload	Load (Project/ Monitoring)	Delete from Monitoring	Controller OPM
save request on that node.						
1-2 A checkpoint restore can step on a save already in progress, as it has no knowledge that a save is occurring. The changing of the controller's database by the restore at the same time of saving will most likely abort the save. The restore can only be done after the controller is placed into IDLE, and then the node and its EE will be deleted. The deletion of the EE will cause communication errors to be seen by the in-progress save. Depending on how long the save is, this either aborts it altogether (if it has not yet saved the EE), or causes the underlying CMs to be marked as stale in the saved checkpoint file due to communication errors on their accesses.						
1-3 Before the save gets initiated, it will make a copy of the checkpoint base files into a working set. If a rebuild/upload comes in at the time that the working set is being used then both complete without problems. This is because the rebuild/upload knows nothing about the working base, as it is rebuilding the base that the working base was copied from. However, if the save is just finishing but rebuild/upload is not yet done, it is possible that the checkpoint save has errors. This is because both the rebuild/upload and the save want to update the node's latest.cp. The updating of this is controlled by an ERDB lock. If save cannot immediately get the lock, it errors out.						
1-4 Due to the length of some checkpoint saves, checkpoint has chosen not to lock the node on saves. Checkpoint wants to allow the user to perform project loads, monitoring deletes, and OPM during that checkpoint save process. When any project load/delete/OPM occurs, the node's latest.cp gets locked, and Compatibility number gets updated in the base. Furthermore, the load/delete/OPM modifies the latest checkpoint file based on the tagged objects being modified. After the save completes, it determines that the older compatibility number, that was captured when save was initiated, no longer matches that in the latest checkpoint file. At the time, the save completes but this saved checkpoint file is immediately disqualified, and discarded/deleted since it's now incompatible. No file replication will occur on this discarded checkpoint file. Depending on timing of the load/rebuild/delete, another case also here - where the project load is set to waiting due to the ERDB lock on the entity due to In Progress Save. When Save is accessing the Checkpoint base files (either for making the working set / copy back), and if a load is requested at that time, it generates an error message for load that the entity is locked.						
1-5 Depending on timing of the load/upload/rebuild/delete, another case also applies here. The load/upload/rebuild/delete is set to waiting due to the ERDB lock on the entity due to the In Progress Save. When Save is accessing the Checkpoint base files (either for making the working set / copy back), and if a load is requested at that time, it generates an error message for load that the entity is locked. The save will be (ERDB) locking it for a short time so as to update node's latest.cp. If the load/upload/rebuild/delete operations comes in just after that lock gets obtained by save, then the load/upload/rebuild/delete operation may immediately abort since it may not wait for the lock to be released.						
2-1 As stated in Note 1-1, checkpoint restore operation runs without knowledge that a save can be						

Operation In Progress	Action Taken for Given Initiated Operation					
	Checkpoint Save	Checkpoint Restore	Rebuild Checkpoint or Upload	Load (Project/ Monitoring)	Delete from Monitoring	Controller OPM
in progress at the same time. If a restore is in progress, and a save is initiated, that save will need to be initiated when the node is in IDLE. This could only be done for a manual save being initiated. The save will complete successfully, abort or result in stale information depending on whether/when the save encounters communication errors. **3-1** A rebuild checkpoint operation, upload, load, monitor delete or OPM operation always will be modifying the checkpoint base and will have locked the base before they were started. When the save is initiated it will attempt to copy the base into the working base. Since the base will not be fully established, the copy will abort or not be initiated at all. This will cause the save to abort.						

Restoring From Checkpoint

This section provides a procedure for using the Checkpoint User Interface to restore a node to a previous operational state using saved checkpoint files.

ATTENTION

After updating the firmware in a C200 controller, **you must reload the Controller block** from either the **Project** or **Monitoring** tab after the restore. Otherwise, you may not be able to issue commands to the CEE.

Prerequisites:

- You have logged on with a security level that permits you to restore checkpoint files through Control Builder or the Detail Display for the selected node in Station.
- You put the node to be restored in its **IDLE** state unless the node has failed. You can Checkpoint restore a failed node without first putting it into its **IDLE** state.
- You put all hardware modules associated with the hardware node in their **IDLE** states unless the node has failed. For example, you have inactivated all the I/O modules associated with a Controller.

Considerations:

- The checkpoint restore function is independent of an operational Engineering Repository Database (ERDB).
- Conditions that can result in incomplete operation checkpoint data (OCD) are:
  - Control Data Access (CDA) errors on acquiring OCD.
  - Selected checkpoint file is the node's latest.cp just after a project/monitor load and before any checkpoint save has occurred.
- If you use a checkpoint file whose entirety is incomplete, certain tagged objects will be restored with only information from the load . A Warning message will be displayed, if you try to use a checkpoint file that only contains configuration checkpoint data (CCD) for restoration. One should view the Details pane in the Restore UI for further information.
- Do not perform any other load, restore or On-Process Migration operation until the checkpoint restore is completed.
- If a delta flag appears next to a node icon in the **Monitoring** tab after a restore operation, do an **Upload With Contents** operation on the given node.

Restoring Custom Algorithm Block (CAB)

Checkpoint captures CAB configuration parameters during the load of the block instance. If there are multiple instances of a CAB type, only one copy is stored and the same copy is referenced by all instances.

The impact of this means:

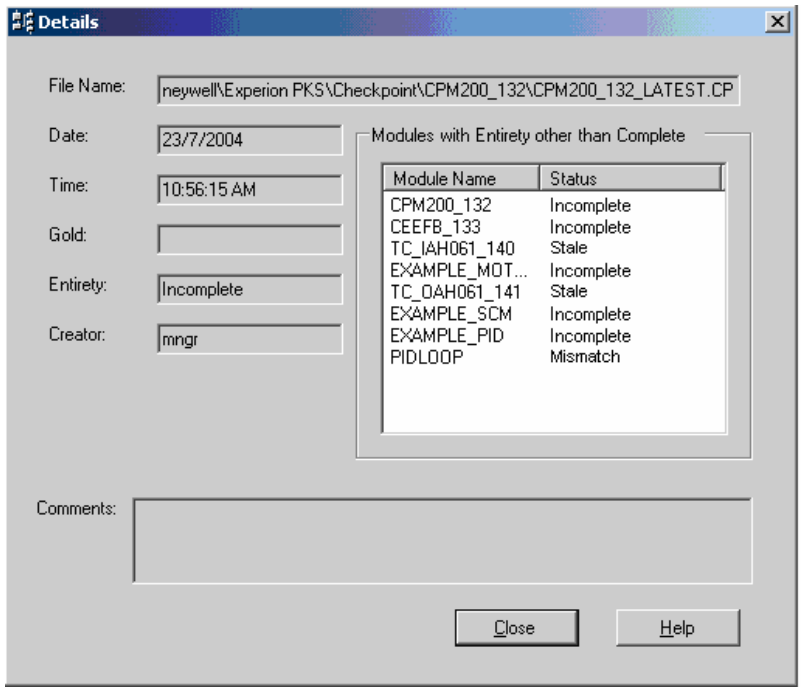
If the block has executed at least once prior to Checkpoint Save	If the block has NOT executed at least once prior to Checkpoint Save
Checkpoint Restore returns the most important initialization seen - NOT the most recent. • Locally defined variable values will not be restored • Custom Data Parameters (CDP)and Parameter Reference (PRef) values will be restored	Checkpoint Restore returns the most recent block.

During a Checkpoint Restore operation, parameters are returned in the same order and format that they were stored during a load operation. The instance of the CAB type's algorithm in the Checkpoint Base structure is deleted, once the only block referencing it is deleted.

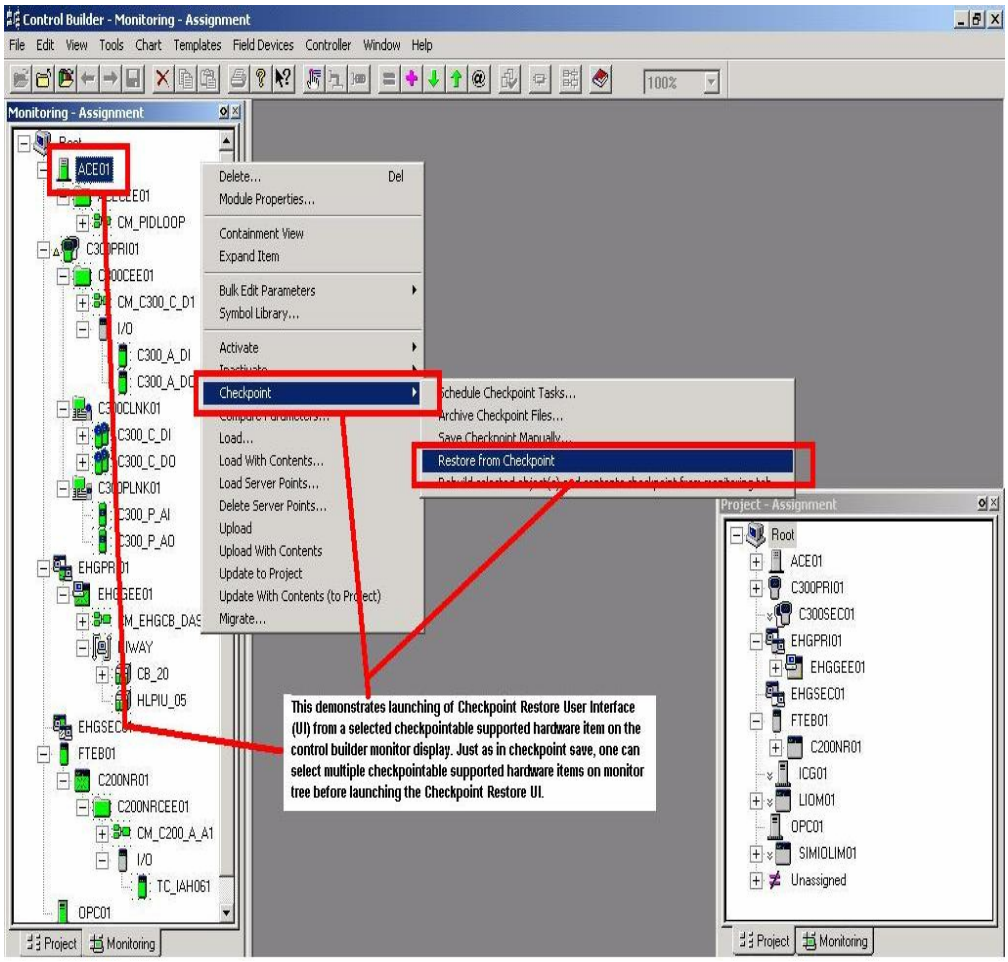
To initiate restore from checkpoint

The following procedure outlines the typical steps for restoring a selected node using the saved checkpoint files.

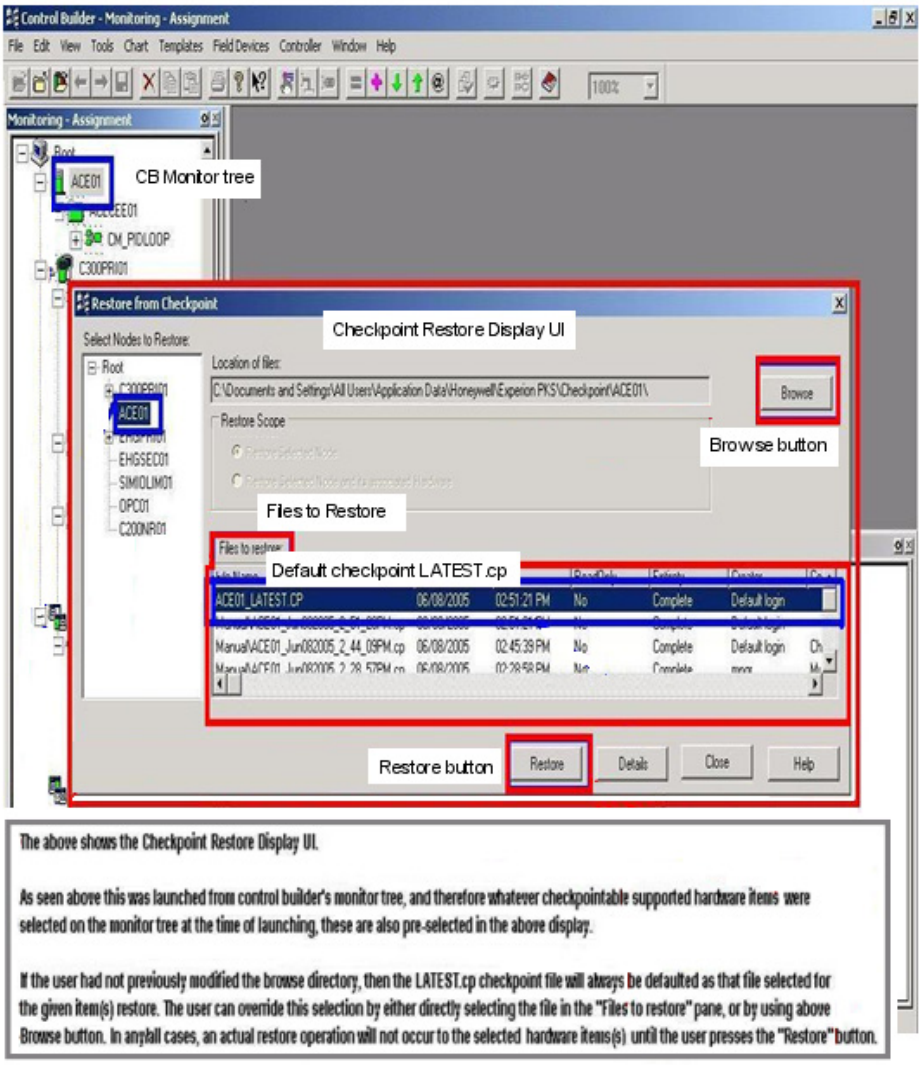
Step	Action
1	On the **Controller** menu, click **Checkpoint** > **Restore from Checkpoint**.
2	In the **Select Nodes to Restore** box, click the ⊞ plus sign to expand the root directory, select the node or nodes you want to restore. Note that selecting two or more nodes will make the checkpoint files listed in the **Files to restore** box unavailable. A message appears stating that each node's latest checkpoint will be used.
3	In the **Location of files** field, be sure directory path and checkpoint file name are correct for selected node or nodes. Use the **Browse** button to select another directory, if applicable. If you have selected two or more nodes, the default checkpoint file name will be for the latest.cp file.
4	If you selected a node with hardware modules, select the check box under **Restore Scope** to restore the node and its associated hardware modules. Leave the check box unchecked, if you do not want to restore the node's associated hardware modules.
5	In the **Select nodes to restore** box, select the node or nodes you want to restore. Note that selecting two or more nodes for restoration will make the checkpoint files listed in the **Files to restore** box unavailable, since each node's latest.cp file will be automatically chosen for the restore
6	When the checkpoint has an entirety state of other than **Complete**, click the **Details** button to determine which tagged blocks this applies to.

Step	Action
	
7	• On the **Details** dialog, you can view the incomplete/stale modules or mismatched ones in the selected file. • Click the **Close** button to close the dialog.
8	Click the **Restore** button.
9	Wait for the restore operation to complete. A **Restore from Checkpoint** dialog appears that will display any errors encountered during the operation.
10	Repeat Steps 2 to 9 to restore other nodes, as required.
11	Click the **Close** button to close the dialog.
12	If you want to keep the Monitoring ERDB of this controller consistent with that just restored to the controller, initiate an **Upload** operation from the Controller to the Monitoring Engineering Repository Database (ERDB). See the *Using Upload command* section in the *Control Building Guide* for more information.
13	This completes the procedure.

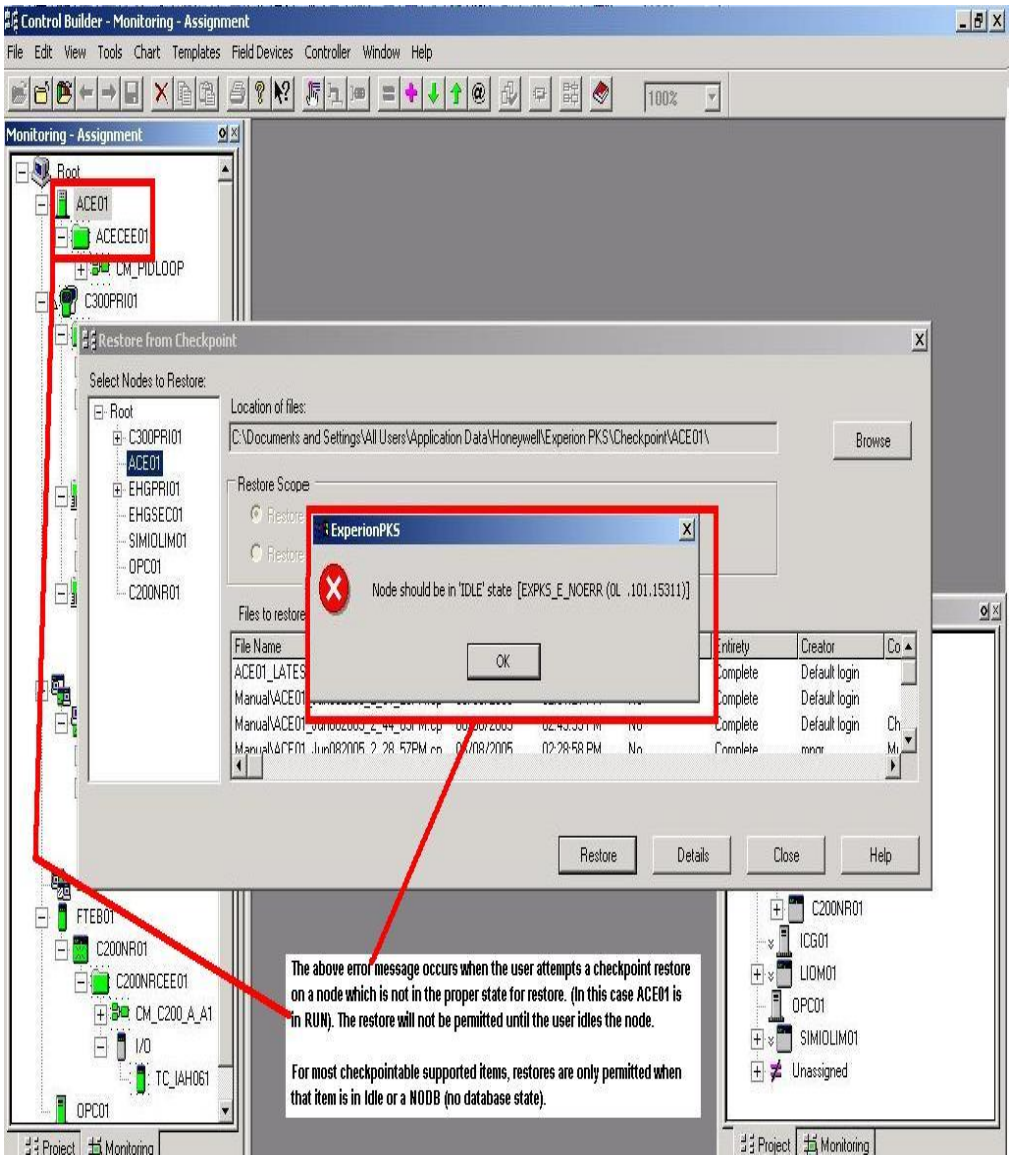
Restore from checkpoint graphical reference



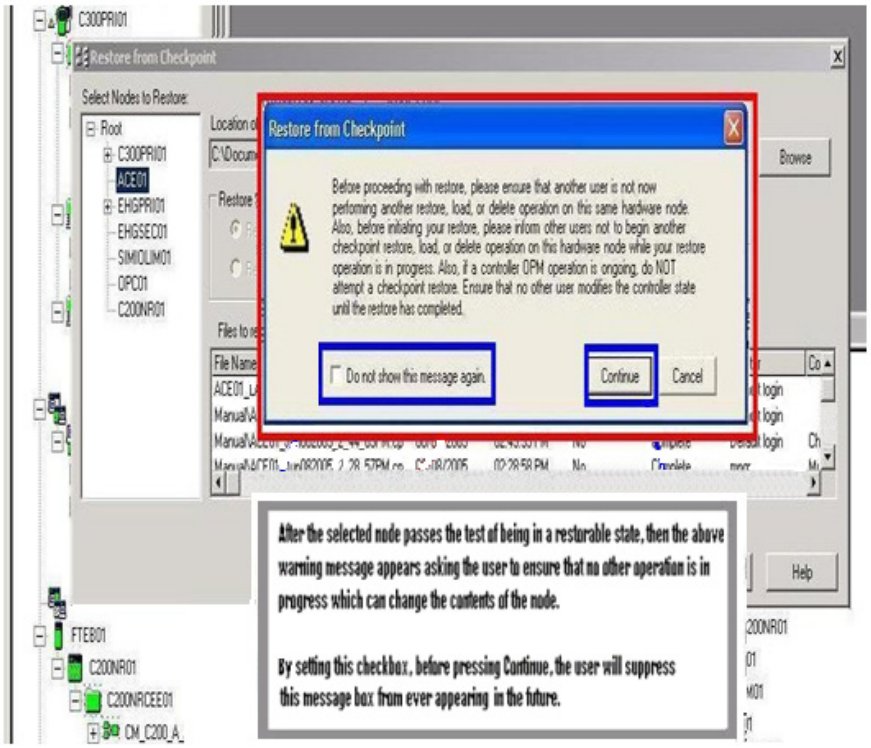
Launching Restore from Checkpoint by right-clicking selected node and selecting from list - multiple node selections are possible



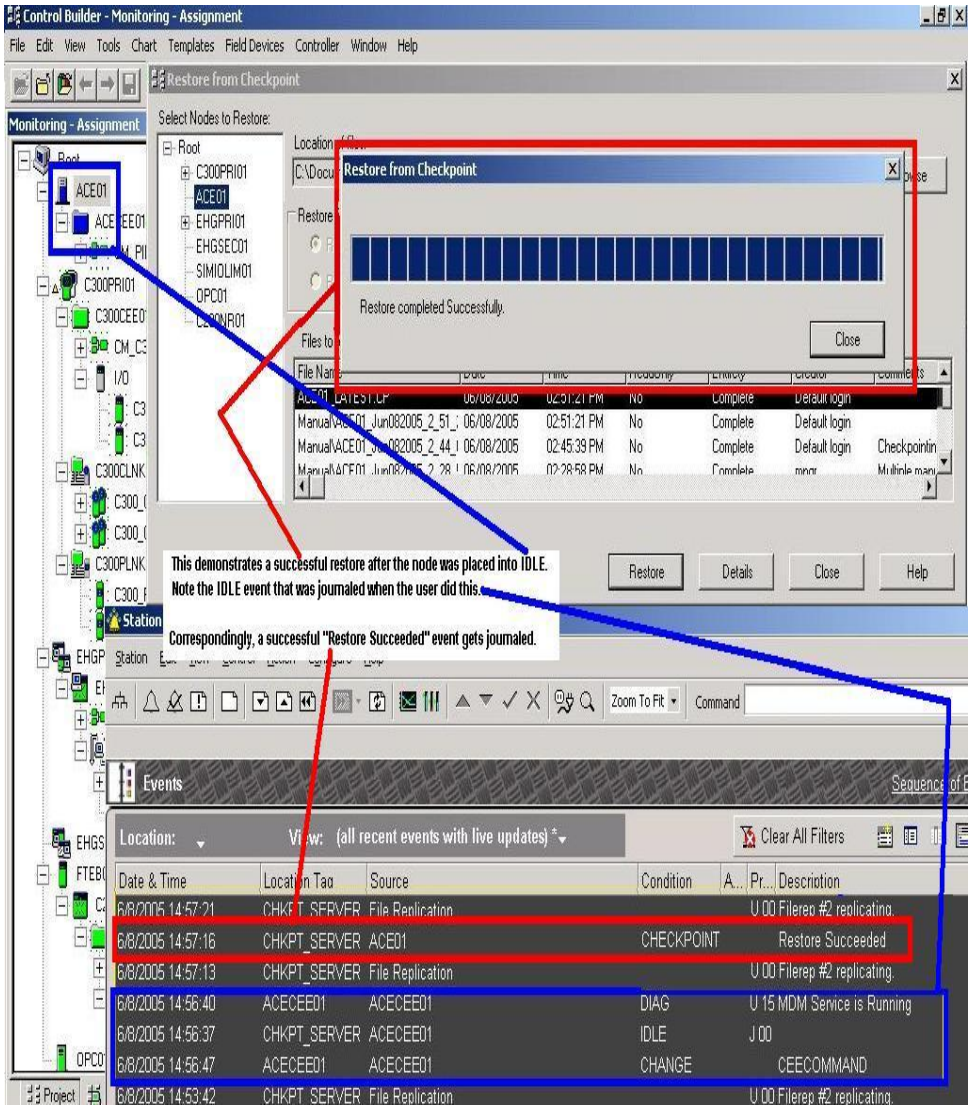
Pre-selecting nodes before launch with default to latest.cp file selection



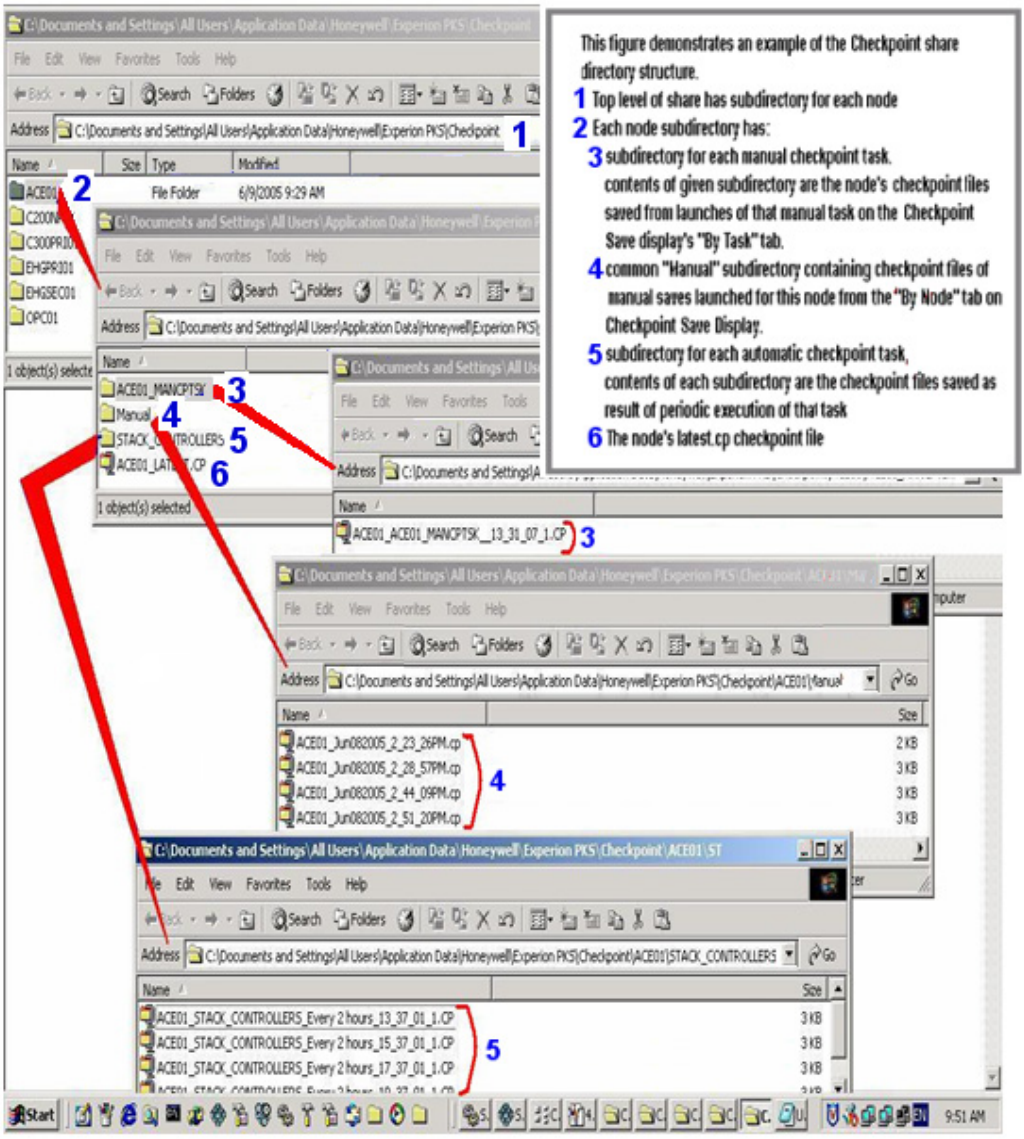
Initiating restore to node in improper state results in an error message



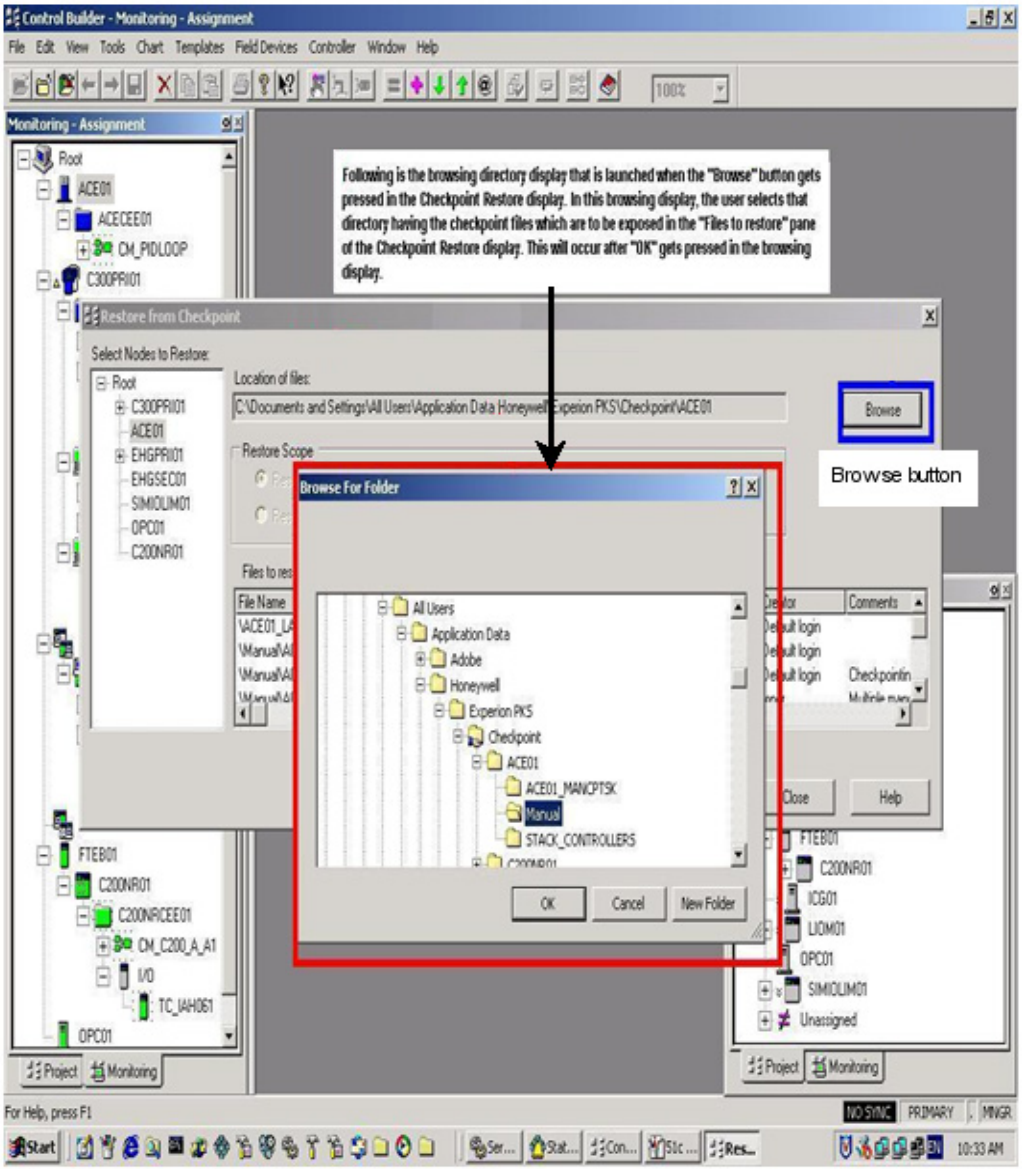
Initiating restore to node in proper state results in warning message to check for conflicting operations before continuing - users can choose to suppress the warning message



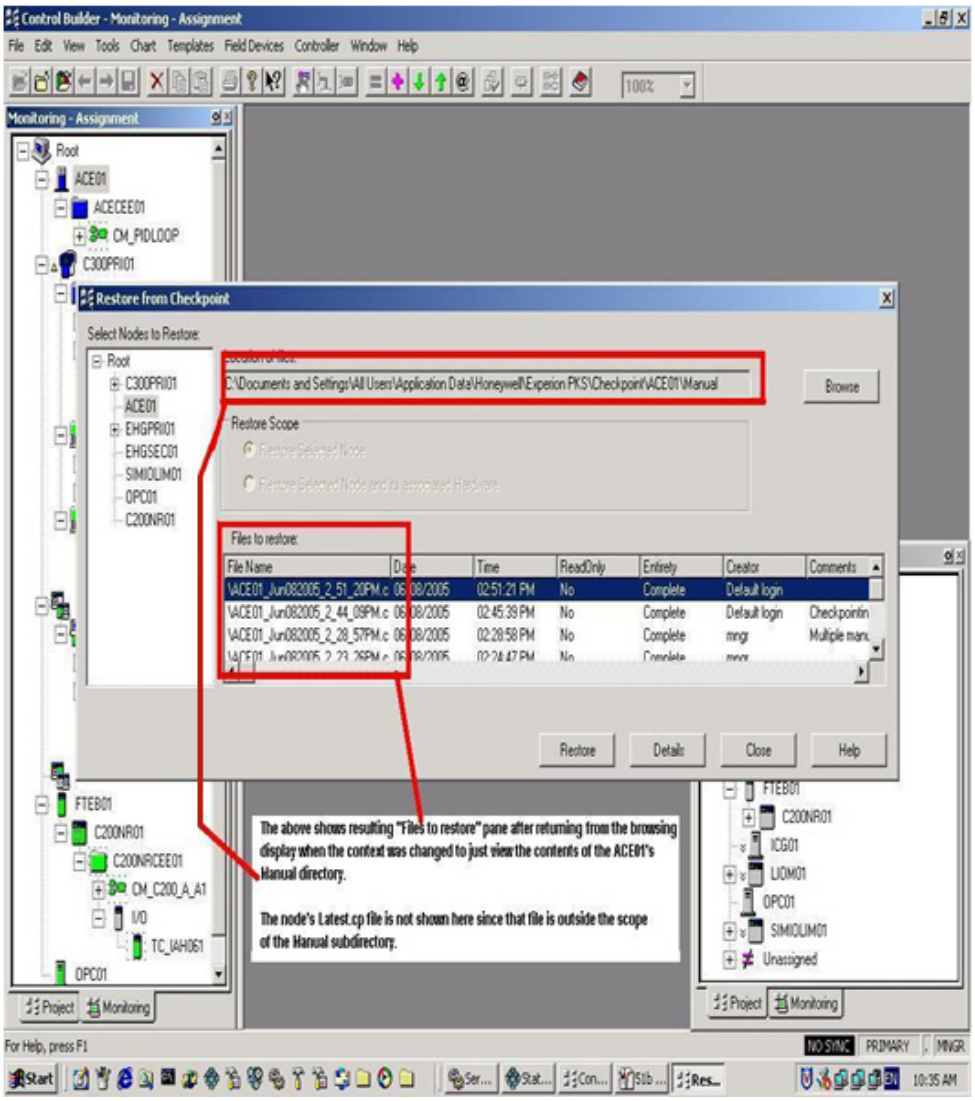
Users can monitor progress of restore operation and check for related journaled events



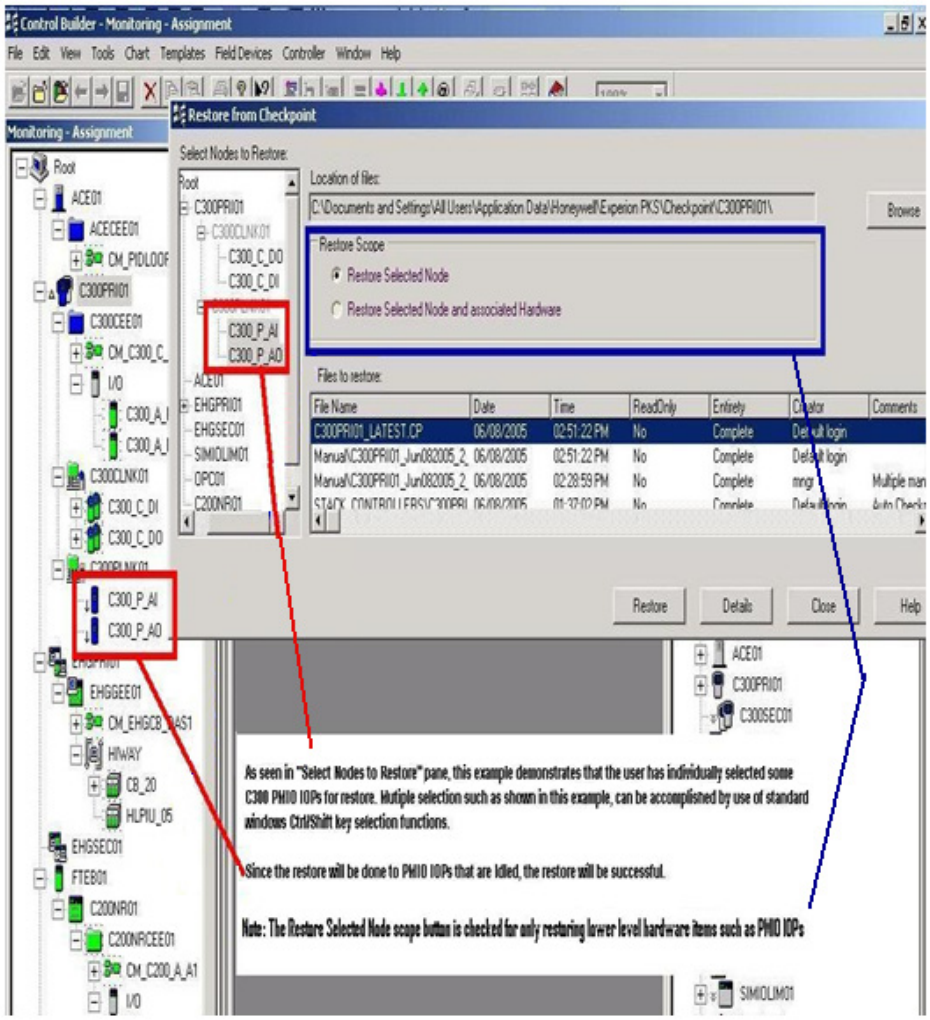
Example of default directories and files



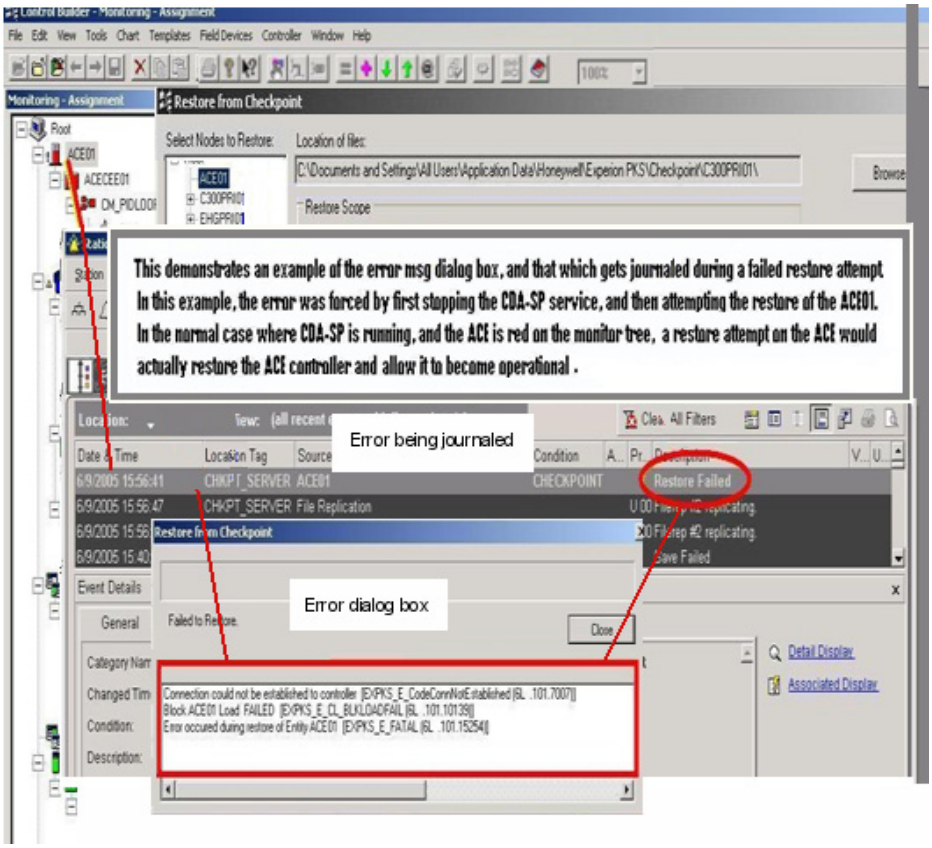
Using Browse button to navigate directories



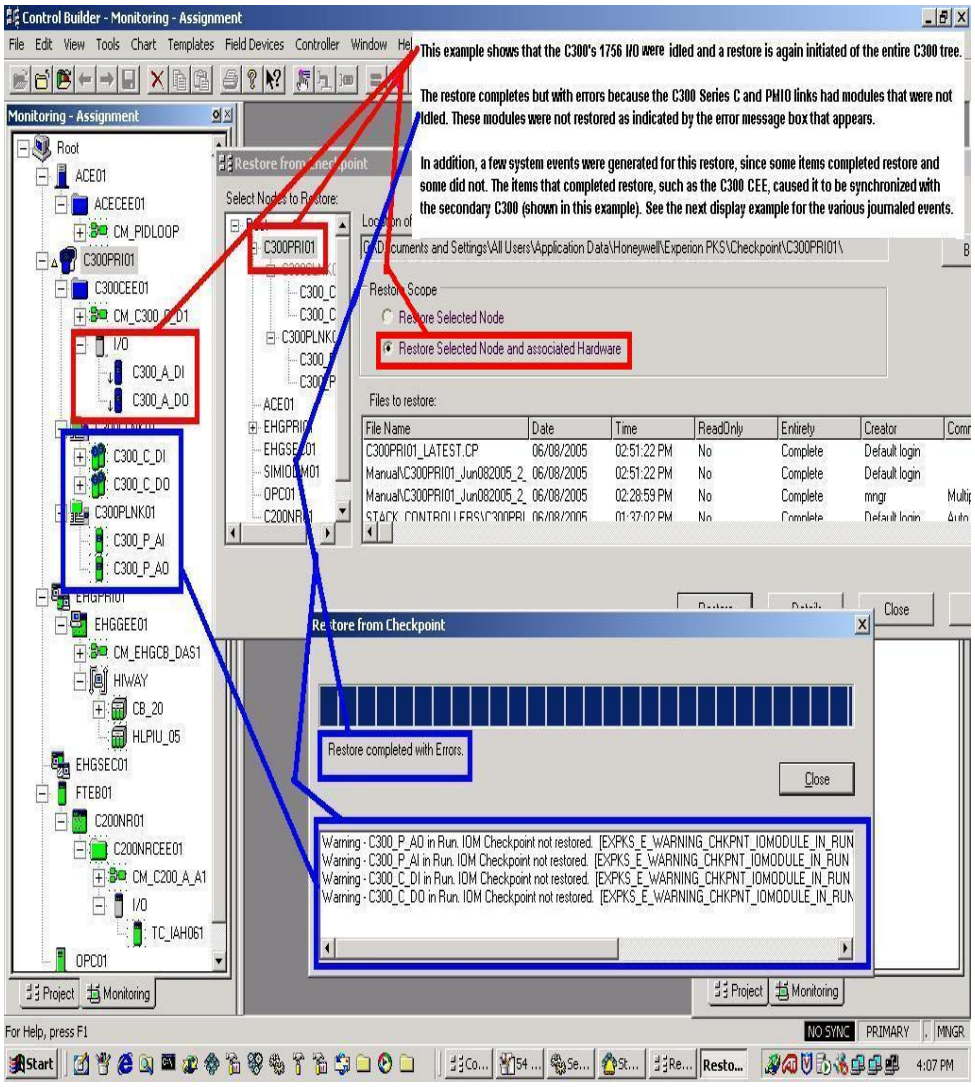
Selected directory location determines what files are available for restore



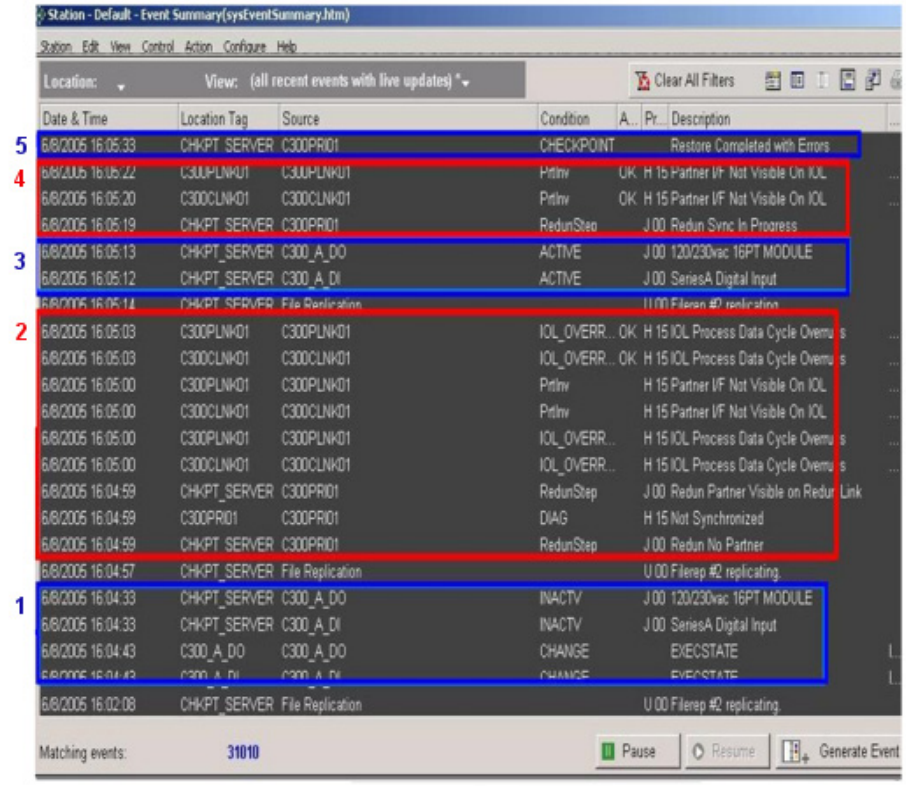
Users can choose scope of restore for top level node to include associated child hardware



Example of error and event notifications for failed restore to top level node and its hardware



Example of restore completed with warnings to top level node and its hardware



The above is an example of the various events that were journaled in the C300 full scope restore that completed with errors as shown in the previous display

1 These are the events associated when the user placed the C300 1756 I/O into Idle so that the restore could then be attempted.

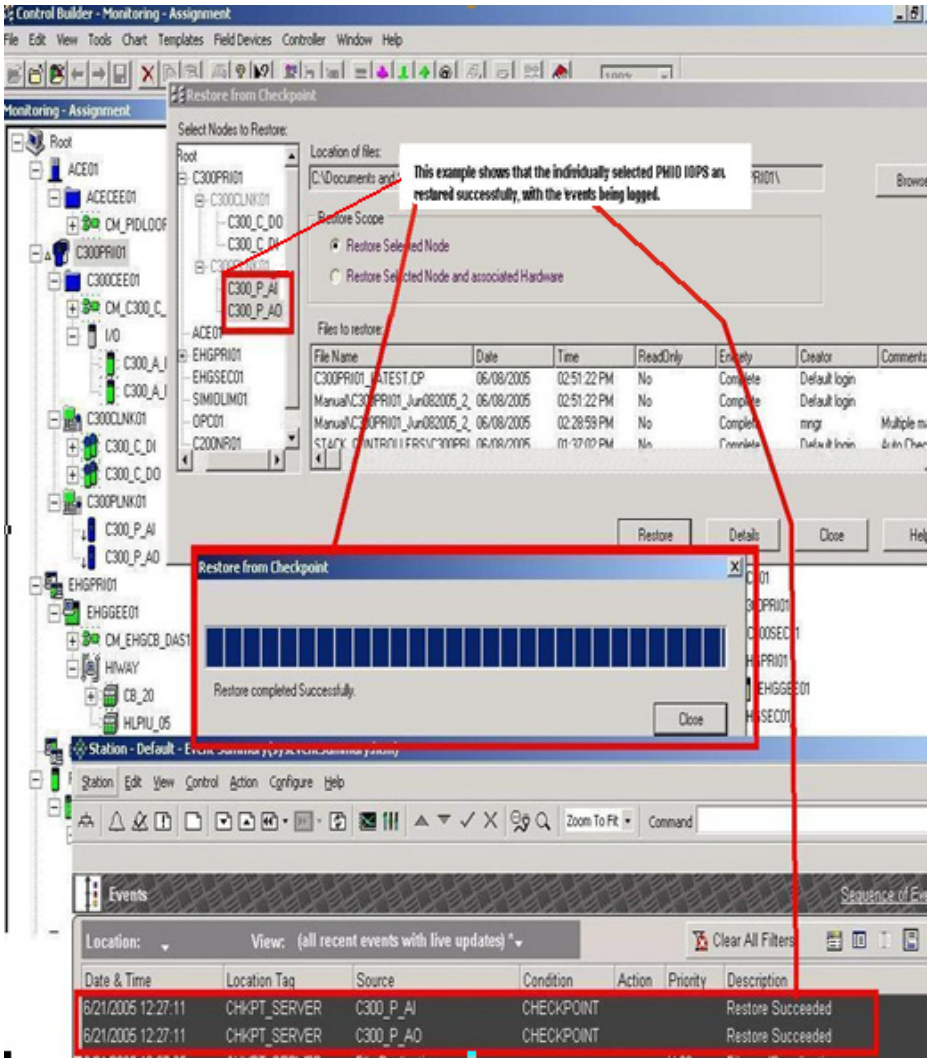
2 These events are caused once the full scope restore started on the C300. Because the primary partner of the C300 is being restored, the Not Sync events appear. In addition, some I/O link events occur since the Series C and PMIO Link EEs are restored (even thougth the underlying modules still in run are not restored).

3 These events are associated with the 1756 I/O that gets restored to an active state since that is how they were originally saved in the checkpoint file.

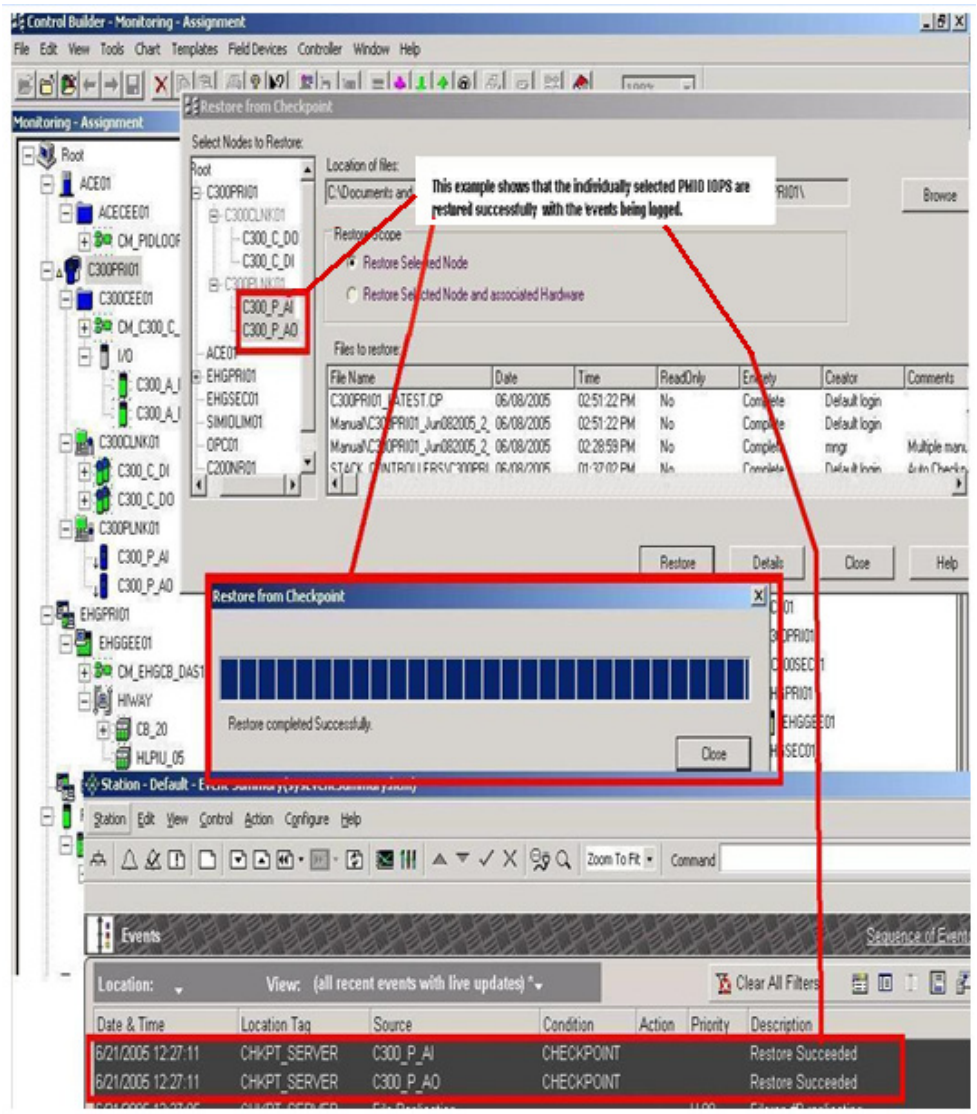
4 As the C300 completes restore, some further return to normal events are journaled regarding I/O Link EEs, and C300 redundancy.

5 Finally, the checkpoint system logs an event stating that the restore completed with errors.

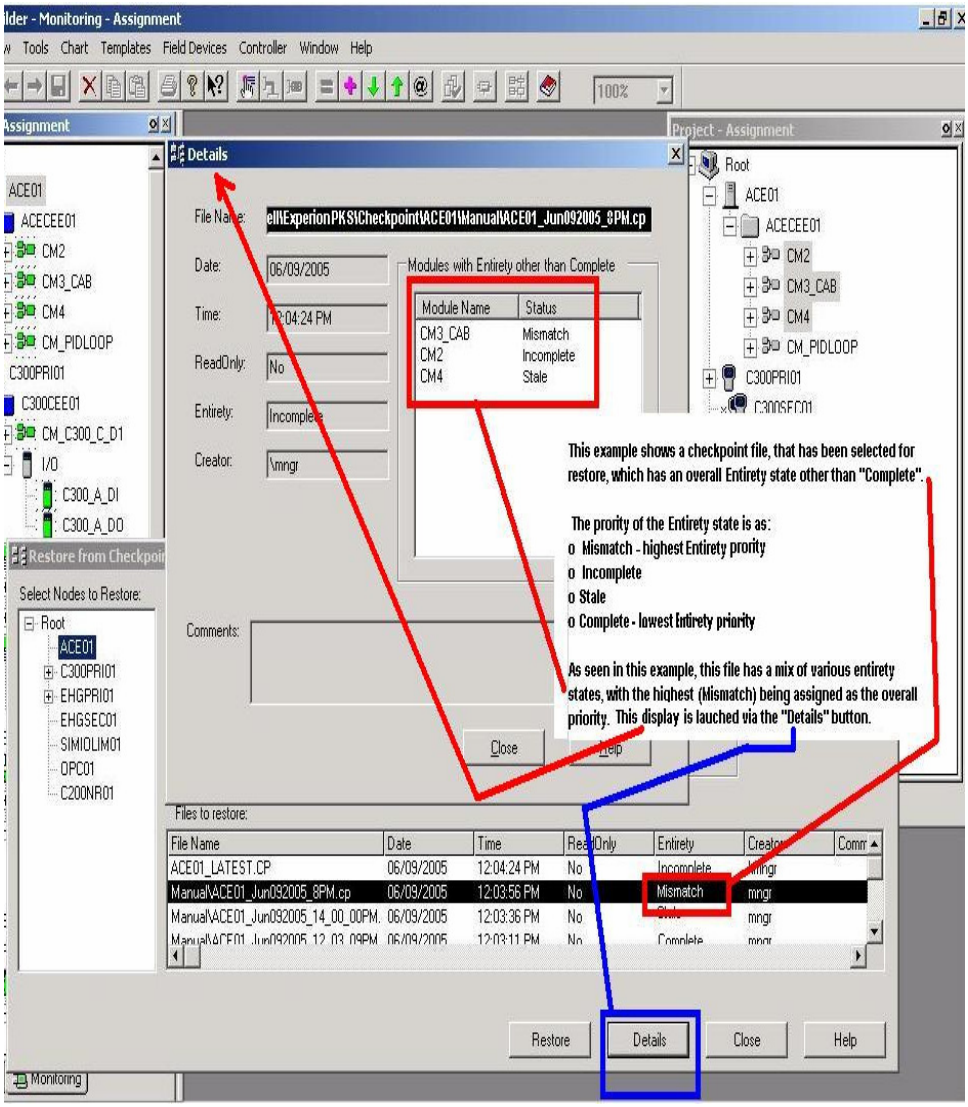
Example of journaled events for restore completed with warnings to top level node and its hardware



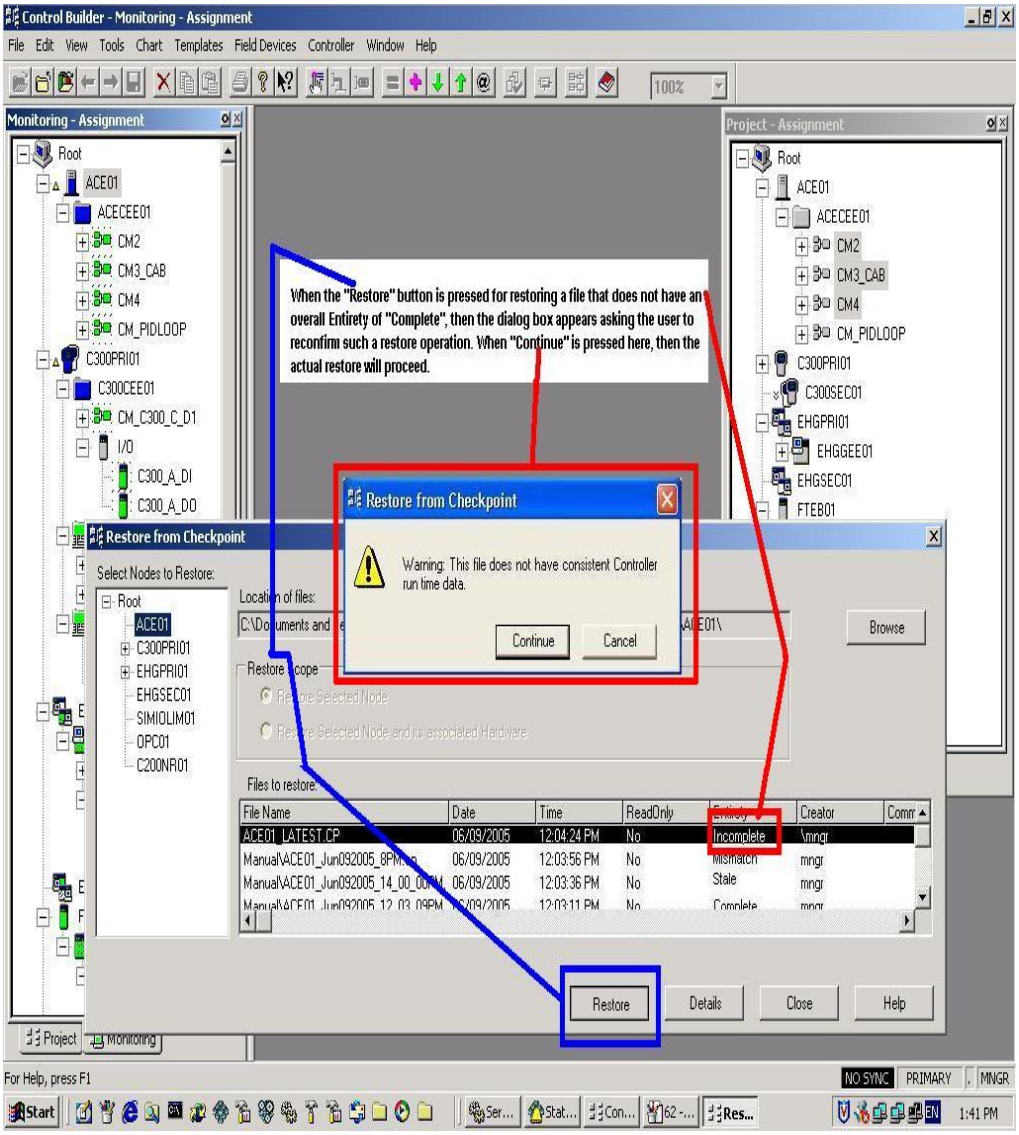
Selecting child hardware for restoration



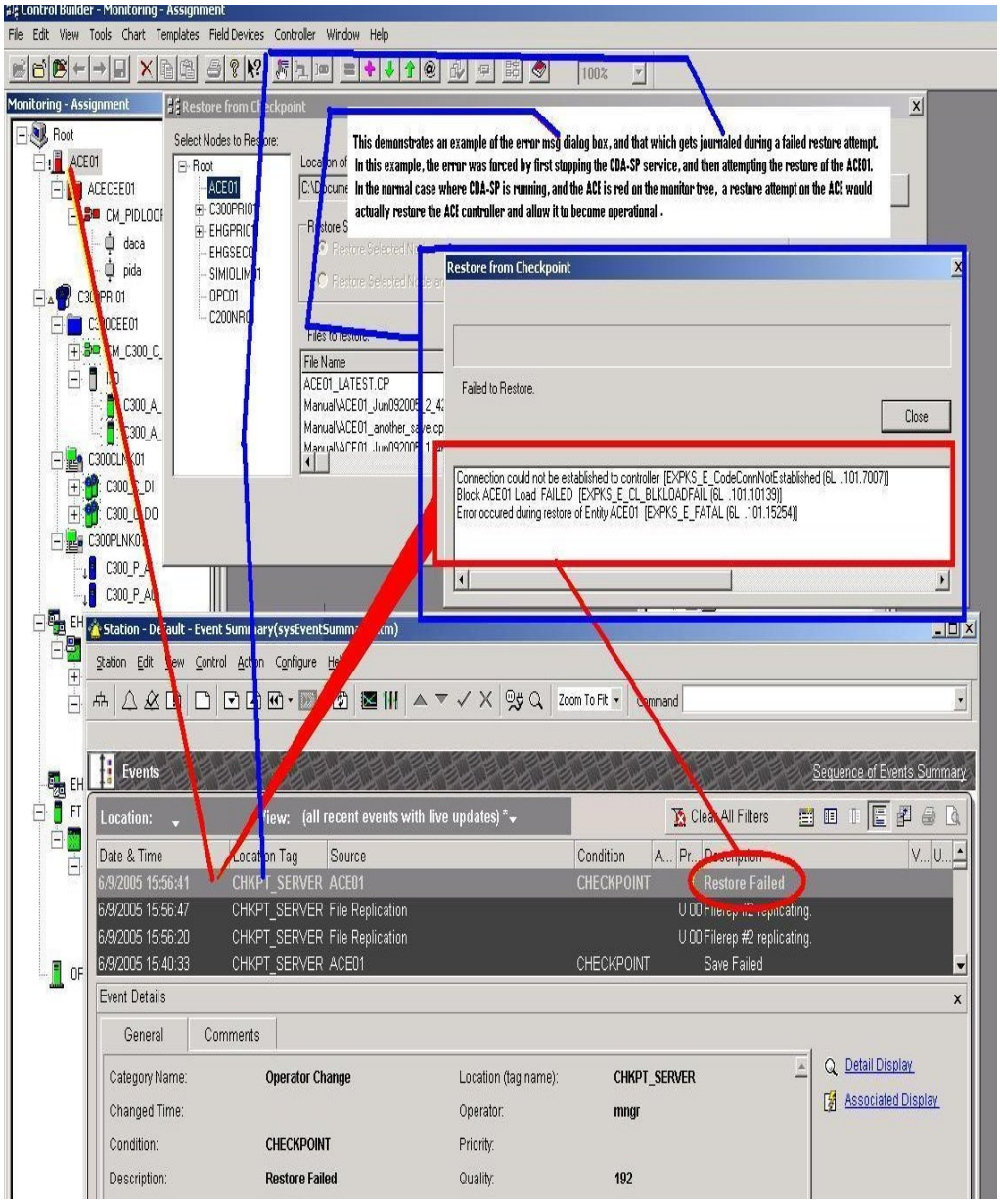
Example of successful restore for Process Manager I/O modules with journaled events



Example of Details dialog for checkpoint file with entirety other than <u>complete</u>



Example of warning message for restoration attempt using a checkpoint file that does not have a complete entirety



Example of restore failure and journaled events when communication link is broken with selected Node

Rebuilding Checkpoint from Monitoring Tab

This section provides a procedure for using the Checkpoint User Interface to rebuild the checkpoint base file as the latest version for the node. This includes rebuilding the node's latest.cp checkpoint file that results in an entirety state of **Incomplete**. This operation is equivalent to re-loading the node contents from the **Monitoring** tab without the information actually being loaded to the controller or server.



ATTENTION

Typically, you do not have a need to perform checkpoint rebuilds. However, if such an activity is done on a node, you must be aware that a **Mismatch** entirety state can be placed on all future checkpoint files. This **Mismatch** condition exists only when custom blocks are associated with the node and is only corrected through one or more subsequent loads. For more information about the entirety attribute mismatch state, see the preceding Entirety Attribute section.

Prerequisites:

- You have logged on with a security level that permits you to rebuild checkpoint files through Control Builder.
- You have saved checkpoint files manually and/or automatically

Considerations:

- After updating the firmware in a C200 controller, you must reload the CPM200 block from either the **Project** or **Monitoring** tab after the restore. Otherwise, you may not be able to issue commands to the CEE.
- Before initiating this function, it is a good practice to do an upload from the controller to monitoring ERDB to have the most recent controller information used for the rebuilding. You only need to upload from controller when a subsequent checkpoint rebuild is going to be done. The upload from server function provides no advantage for the checkpoint rebuild.
- The **Rebuild selected objects(s) and contents checkpoint from monitoring tab** function is equivalent to a monitor load operation except that the configuration checkpoint data (CCD) is written to the checkpoint file without actually loading anything to the controller or server. This function is typically only needed for the following scenarios.
  - Used in server OPM, while in dual primary mode just after the former secondary server has been migrated to the new release. The user must manually

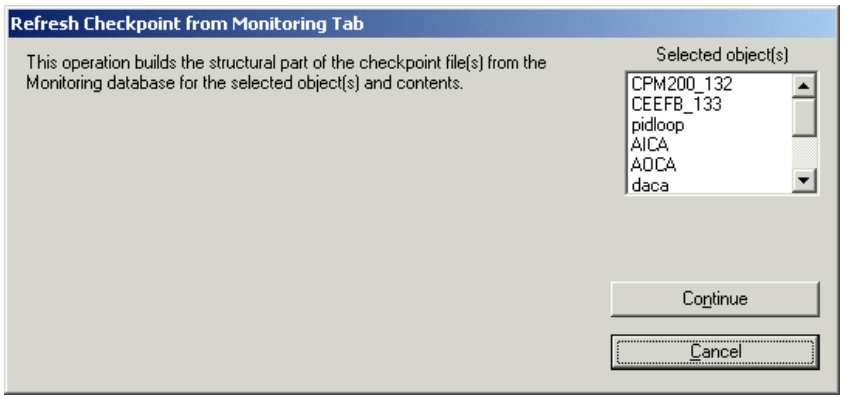
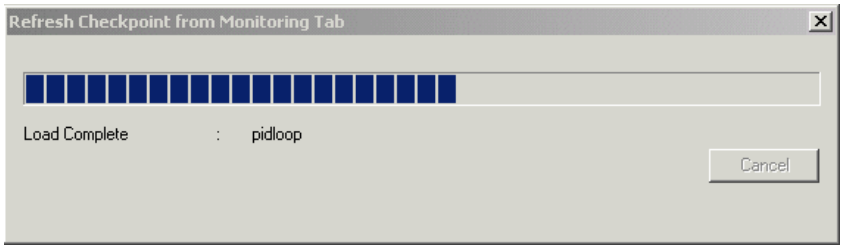
rebuild all checkpoint files on the new release primary and then perform **Manual** saves to get each node's latest.cp to a complete entirety state. This ensures that the user has checkpoint restore coverage in case a controller node fails while the dual primary mode is removed and before the former old primary gets installed with the new release and comes up as a secondary.

- Used by controller OPM wizard when migrating a controller. (This occurs after server migration. It is executed automatically by the OPM Controller Wizard.)
- Used when the node's latest.cp checkpoint file might become physically corrupted due to various reasons, such as disk corruption or through some software anomaly.

- In a rebuild operation, only a subset of the load warnings/errors are reproduced. This subset only consists of those warnings/errors that were sourced by the Control Builder. These appear in the error message reporting area just under the progress bar. The user should ensure that clean loads were done to the node before performing this rebuild. Otherwise, the same load errors will be produced in any future checkpoint restore using those checkpoint files that have been subsequently created/saved from this Checkpoint CCD base information recreated during the rebuild.

To initiate checkpoint rebuild

Step	Action
1	In the **Monitoring** tab, select the node whose checkpoint file needs CCD rebuilt.
2	On the **Controller** menu, click **Checkpoint** > **Rebuild selected objects(s) and contents checkpoint from monitoring tab**.

Step	Action
	
3	On the **Refresh Checkpoint from Monitoring Tab** dialog, confirm that the correct node and all its contents are listed in the Selected object(s) box. If node is **not** correct, click the **Cancel** button and repeat Steps 1 and 2 for the correct node. Otherwise, go to the next Step.
4	Click the **Continue** button to initiate the rebuild function. A **Refresh** progress dialog appears for monitoring the status of the operation. If you want to abort the operation without saving any changes to checkpoint file, click the **Cancel** button. 
5	Wait for the rebuild operation to complete before continuing with other functions. If severe load errors appear in the dialog box, this implies that those tagged blocks are not in the rebuilt checkpoint base, because those blocks could not have been loaded to the controller. The rebuilt checkpoint file is to contain only those blocks that were loaded to the controller as reflected by the present Monitoring view.

Step	Action
6	This completes the procedure.

Using Detail Displays for checkpoint operations

As noted in the previous *Station detail display Checkpoint interface summary* section, you can initiate the following checkpoint operations from the Detail Displays for the selected node and/or for a node's applicable hardware children on a Console or Flex station.

- Save checkpoint manually
- Restore from checkpoint
- View checkpoint tasks status

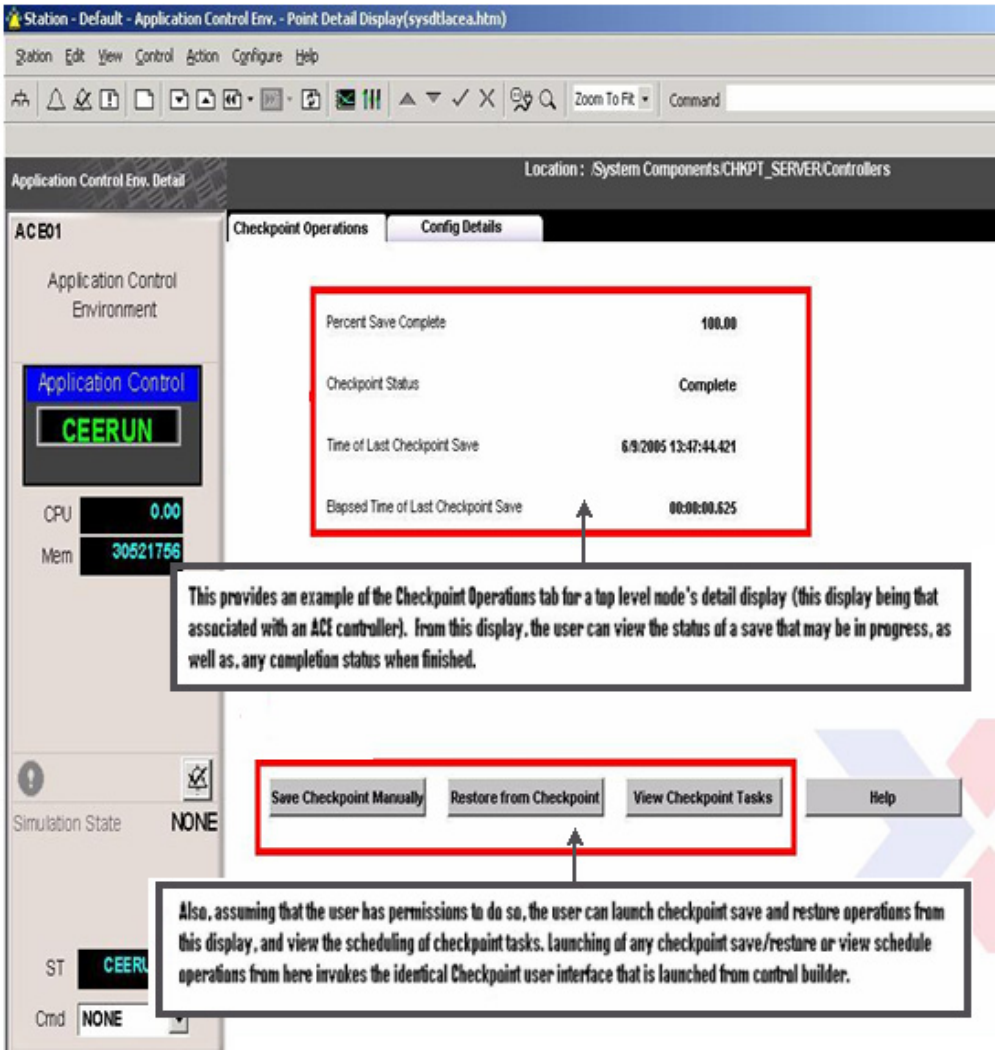
The following table summarizes what operations are available through corresponding buttons on the Detail Display for a given node.

If Node or Child Hardware Is . . .	**Then, This Operation Launch Button Is Available On Detail Display. . .**		
	Save Checkpoint Manually	***Restore from Checkpoint***	***View Checkpoint Tasks***
ACE, C200, C300, ICG, LIOM, OPC, SIM-C200, or SIM-ACE	Yes	Yes	Yes
SIM-IOLIM	SIMIOLIM's Detail Display does not have a **Checkpoint Operations** tab for launching any of these operations. The user must use Control Builder to launch Checkpoint operations for SIMIOLIM.		
C300's - Series C I/O and PM I/O Child Hardware Modules	No	Yes	No
SIMIOLIM's - PM I/O Simulated Child	SIMIOLIM's Detail Display does not have a **Checkpoint Operations** tab for launching any of these operations. Also, checkpoint functionality from Control Builder does **not** provide the ability to perform any of		

	Then, This Operation Launch Button Is Available On Detail Display. . .
Hardware Modules	these checkpoint operations on selected simulated PM I/O modules.

Launching checkpoint operations from Detail Displays

The following illustration and table summarize the checkpoint operations you can launch through selections on the Checkpoint Operations tab of a given node's Detail Display. Once a given operation is launched, it functions in the same way as the corresponding operation does when launched from Control Builder.



ATTENTION

You must have the required permission level to initiate a given operation as previously defined in the *Configuring operation permissions for Checkpoint functions* section.

If You Click This Button . . .	Then, It Launches . . .
Save Checkpoint Manually	The **Save Checkpoint Manually** dialog. See the previous *Identifying Functions on the Save Checkpoint Manually Dialog* section for interface details.
Restore from Checkpoint	The **Restore from Checkpoint** dialog. See the previous *Identifying Functions on the Restore from Checkpoint Dialog* section for interface details.
View Checkpoint Tasks	The **Checkpoint Scheduler** dialog with **Tasks by node** tab showing and the given node pre-selected in the **Available** list box. See the previous *Identifying Functions on the Checkpoint Scheduler Dialog* section for interface details. If you click the **Tasks** tab, the **New**, **Edit**, and **Delete** buttons are not available, since the **Define Task** dialog cannot be launched using this launch scenario.

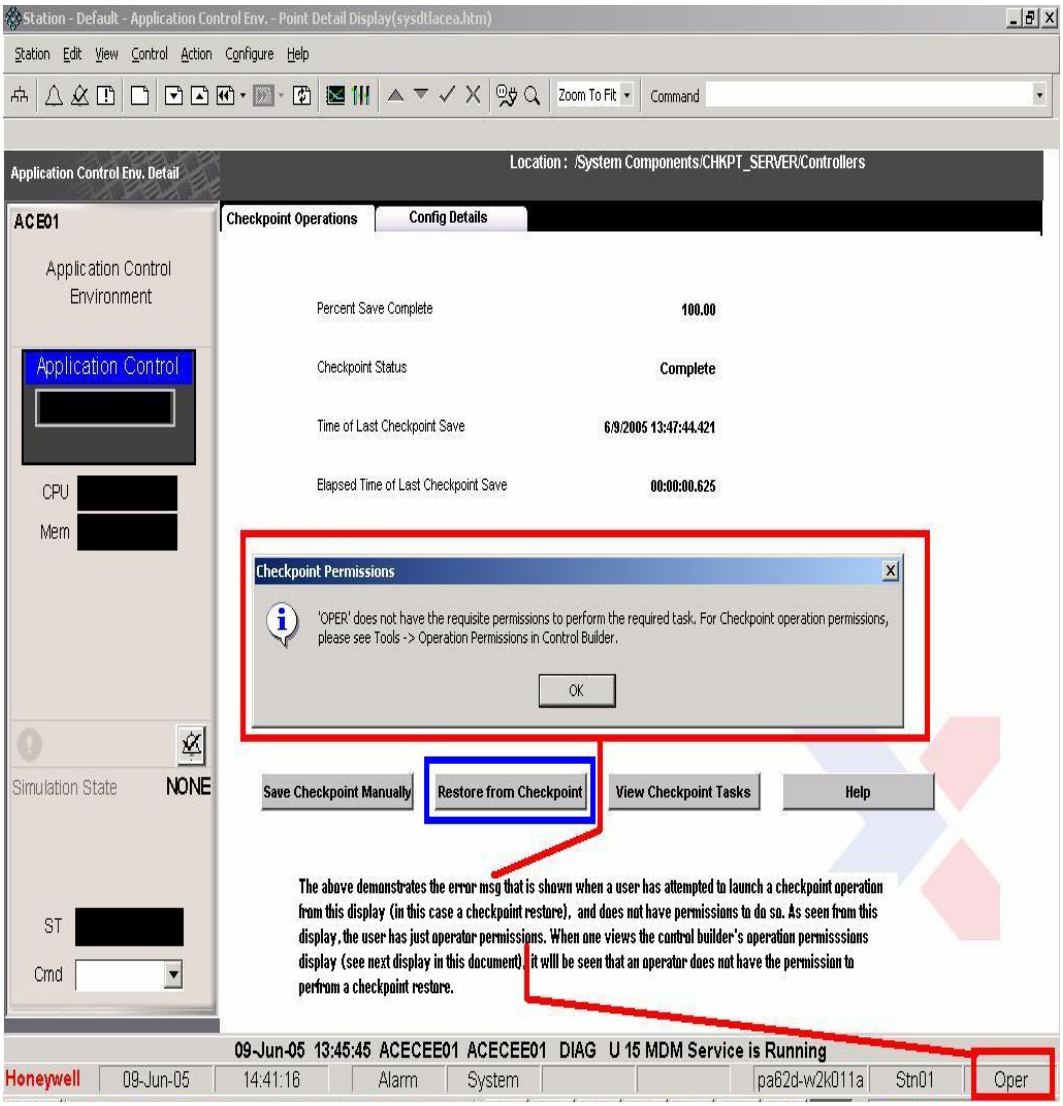
Viewing parameters on Detail Displays

You can view the following checkpoint save related parameters on the **Checkpoint Operations** tab of Detail Displays that include the **Save Checkpoint Manually** button.

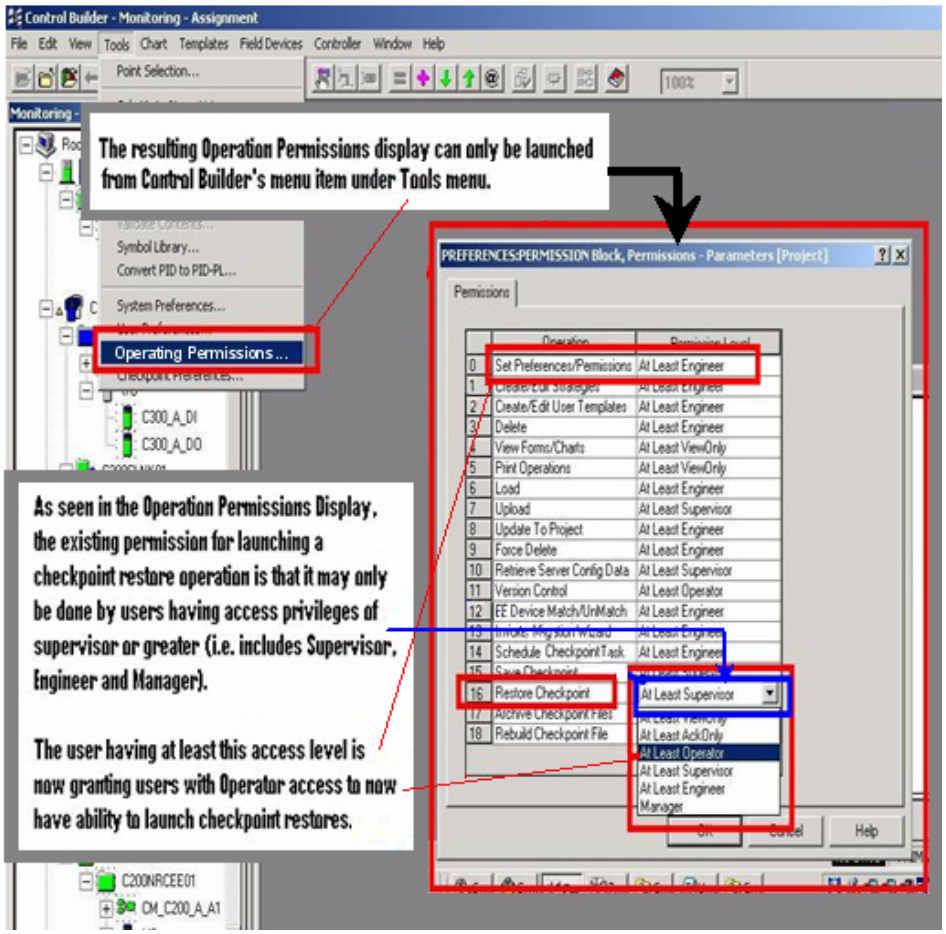
Parameter Name	Description
CPSAVCOMP PERC	Percent Save Complete - Displays progress completion as a number representing the percentage (0-100) of how much of the node's save has been completed. This value gets updated on the node's detail display Checkpoint Operations tab every 30 seconds as the checkpoint save is progressing.
CPSTATUS	Checkpoint Status - Shows the state of the node's checkpoint save. It has the following enumerated values: • Running • Complete • Complete with Stale • Failed
CPLASTSAVE	Time of Last Checkpoint Save - Shows the primary server's wall time/date of when the last checkpoint save was started on this node. If CPSTATUS is Running, then it is the start time of the save that is in progress.

Parameter Name	Description
CPTMEOFSAVE	Elapsed Time of Last Checkpoint Save - Shows the time that it had taken to perform the last checkpoint save on this node. It is expressed in hrs:mins:sec:millisecs . The wall time of when the last checkpoint save actually completed would be this elapsed CPTMEOFSAVE added to CPLASTSAVE.

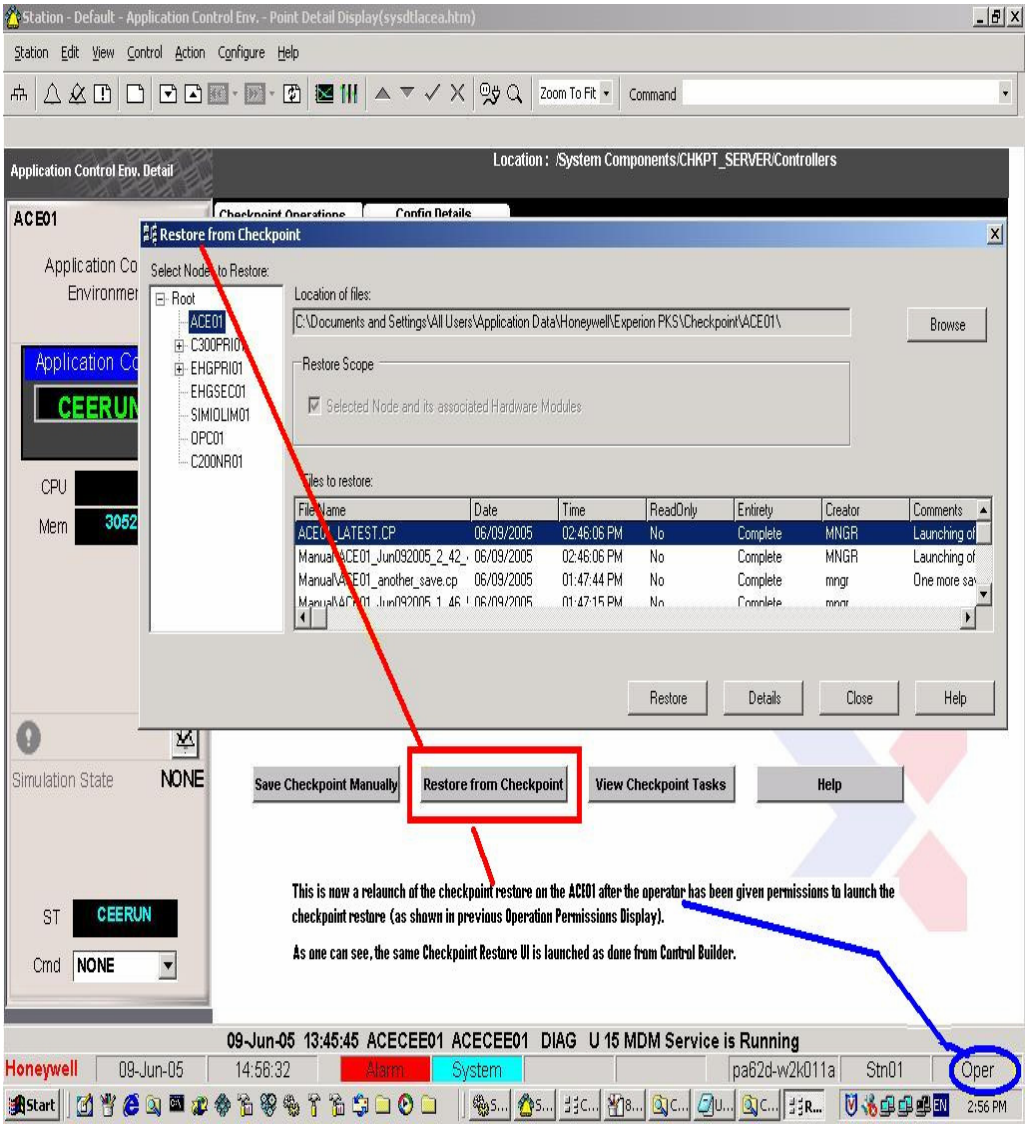
Checkpoint operations through Detail Displays graphical reference



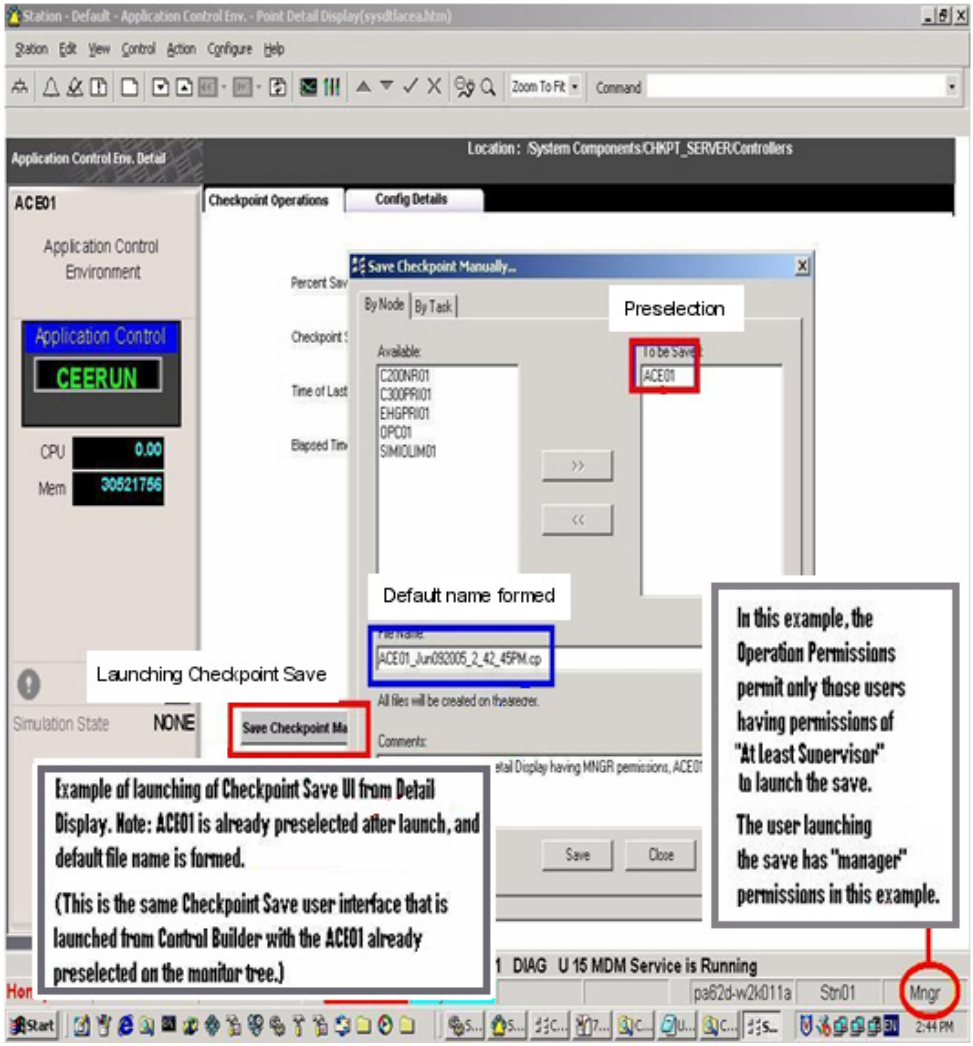
User needs required permission level to launch checkpoint operation as defined through Control Builder Operation Permissions



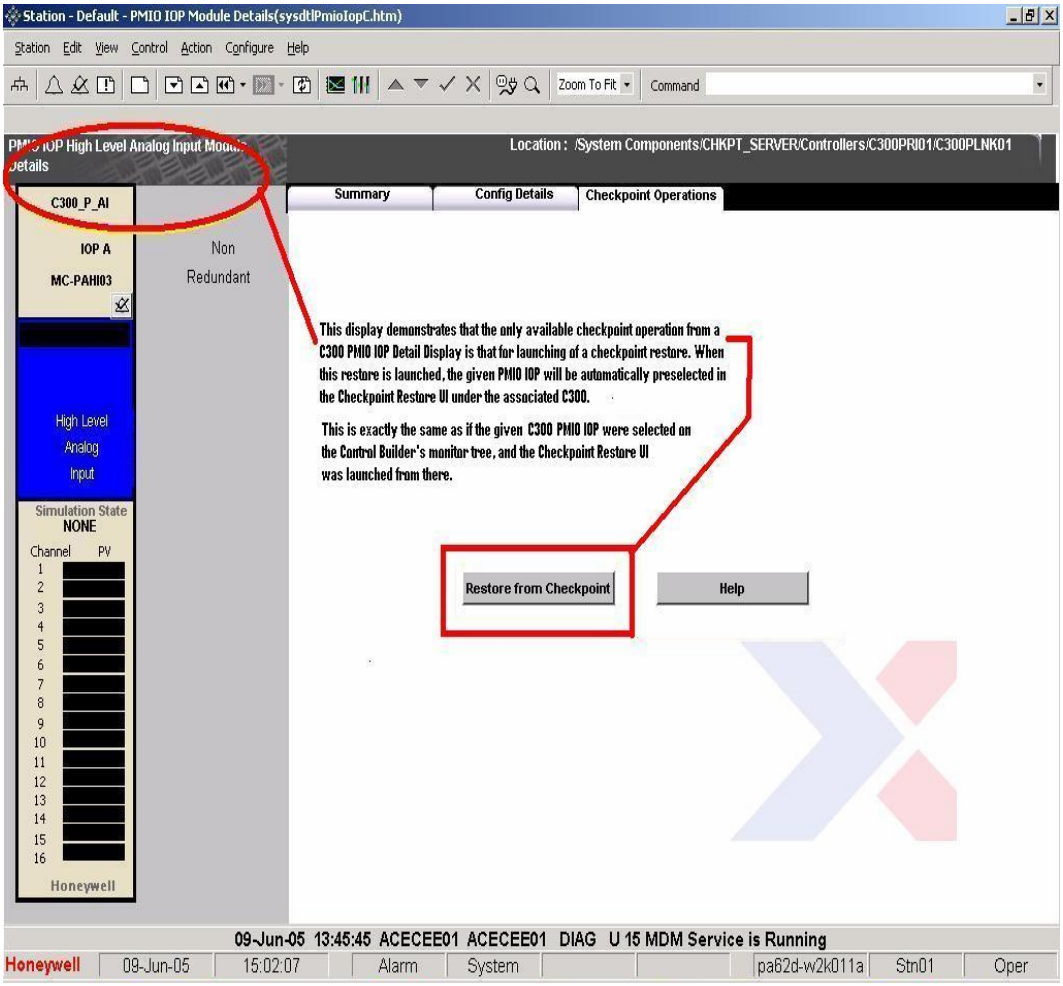
Setting user permission levels through Control Builder Operation Permissions



User can launch checkpoint operation when permission level matches or is greater than one defined through Control Builder Operation Permissions



Launching manual checkpoint save shows pre-selected node as to be saved



Example of Checkpoint Operations tab for C300 child hardware Series C I/O or PM I/O module Detail Display

Troubleshooting Checkpoint Function

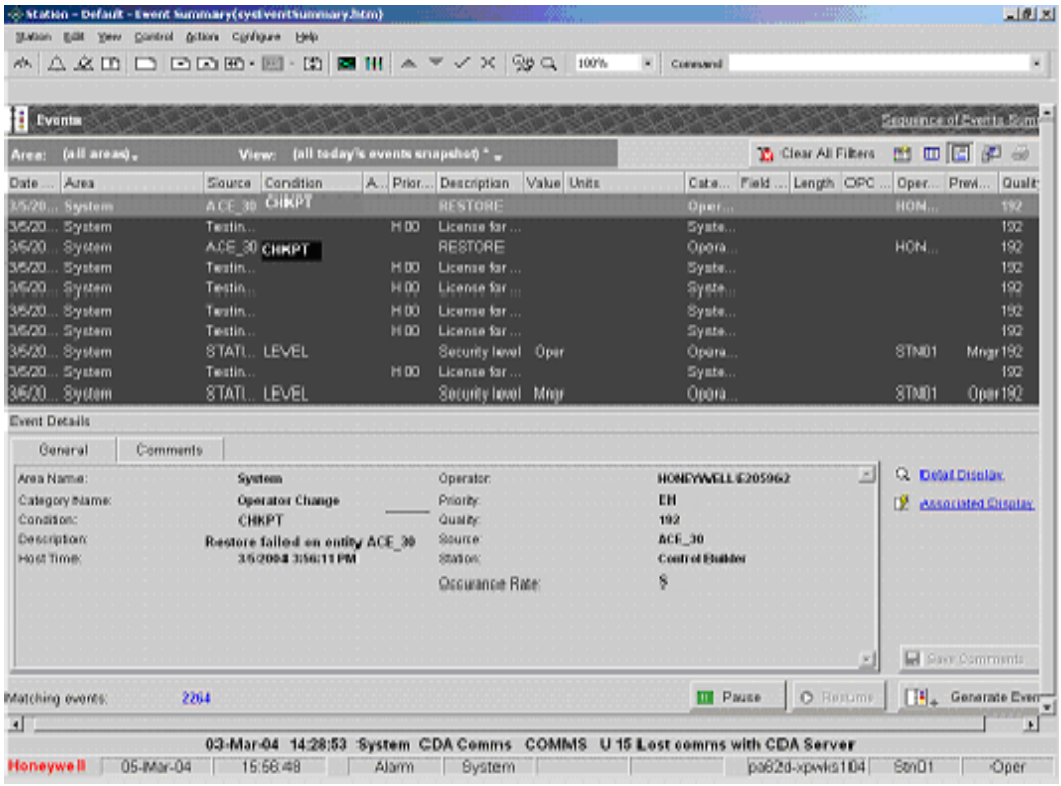
An indication of a problem may be in the form of an error dialog box that includes an error message and possibly an error code, or an event message in the Event Summary display on Station.

Please refer to the *Control Builder Error Codes Reference* book for applicable error code information. The syntax for a typical Control Builder error message is as follows:

Module checkpoint is incompatible EPKS_E_CL_CMINCOMPATIBLE(6L.101.10631)

In this syntax, the error code is the last five digits in the message or **10631**.

The event messages are self-explanatory and include pertinent information in the Event Details display as shown in the following illustration.



Typical Checkpoint event message in Event Summary display in Station showing event details.

Viewing error logs

You can view the Experion error log files stored in this default location on the primary server.

```
>C:\Documents and Settings\All Users\Application
Data\Honeywell\Experion PKS<
```

The default name for the Experion error log file is ErrLog_nn.txt where nn is a sequential number beginning at **1, ending at 10**. Once an error log file reaches the size of 1,001 KB, it is closed and the next error log file in sequence is created to receive the next logging of errors. Once ErrLog_10.txt is completely filled, the process wraps around and begins using ErrLog_1.txt again. All prior information is discarded when the error log file begins to be reused for new errors.

Identifying checkpoint errors logged but not journaled

The following table identifies errors that are logged in the Experion error log, but are not journaled as events in the Event Summary display on Station. Some of these errors do generate more generic event counterparts that do get journaled as described in the following *Checking messages journaled as events* section.

If Error Message Is . . .	Then, Possible Cause is . . .
Undefined Error	Any unspecified error occurred
Periodic Checkpoint Save Failed - Entity *SomeName* not accessible	Auto save failed for any entity due to communication problems with the controller
Manual Checkpoint Save Failed - Entity *SomeName* not accessible	Manual save failed for any entity due to communication problems with the controller
Checkpoint Restore Failed - Entity *SomeName* not accessible	Restore failed for an entity due to communication problems with the controller.
Checkpoint Restore Not Permitted - Checkpoint file for the entity *SomeName* is corrupted	The checkpoint file selected for restore did not pass CRC check. It is physically corrupted and the restore did not continue.
Checkpoint Restore Not Permitted for entity *SomeName*. Target block is incompatible with the block in checkpoint file	The checkpoint file selected for restore of an entity is having incompatible information.
Failed to write summary tab Information	Summary tab of the file cannot be written due

If Error Message Is . . .	Then, Possible Cause is . . .
	to error of access problems
Failed to read summary tab Information	Summary tab of the file cannot be read due to error of access problems
File compression failed	Creation of checkpoint file failed
File extraction failed	Extraction of Checkpoint file failed
File Corrupted - CRC Mismatch	CRC check selected file failed. Select another file.
Invalid Path	Specified path does not exist for the selection.
Creation of Checkpoint attribute file failed	The attribute file creation failed.
Failed to open the checkpoint file for the entity *SomeName.cp*	The checkpoint file could not be opened for read/write.
The list of TLHPE names could not be obtained from SR	Top level node names could not be read from SR. Probably due to SR SYNC issues or connection failure.
Couldn't access CtrlSession	Create, Edit, Delete of task failed, as the ERDB access is not available.
Could not access Sysrep for information	Operation failed. Could not access Sysrep for information.
Failed to connect to Server	Failed to Connect to Server.
Failed to disconnect from server	Failed to Disconnect from Server.
Task Name cannot have invalid characters	Invalid characters are entered in the task name while defining or editing the task in the define task dialog.
Task name exceeds the maximum length of 40 characters	Length of task name exceeds 40 characters in the define task dialog.
Version range is 1-100	User enters the version number which exceeds 100 or which is less than 1 in the define task dialog.
Version should be numeric only	Non numeric version number is entered in the define task dialog.
Resource string not found	A string could not be read from the resource

If Error Message Is . . .	Then, Possible Cause is . . .
	file (for localization).
Grid Initialization failed	The flex grid initialization of Checkpoint interface failed.
Invalid File Name	The checkpoint file name entered for the manual save is not valid.
Node should be in 'IDLE' state	While restoring a node, if it is not in idle state.
Not all nodes are in 'IDLE' state	This can occur when attempting to restore multiple nodes. All of the nodes must be in idle state. One or more of them are found to be in some other state.
Inconsistency is detected. Recommended to Press the Refresh button.	Checkpoint interface receives a message from the service and the taskID does not match the taskID for which the interface had sent a request.
Inconsistency is detected with this task. Recommended to delete this task.	This can occur while editing a task. The Delete task from NTScheduler is successful but Add task to NTScheduler fails.
Failed to create New task.	Checkpoint interface sends a request to the service for creating a new task and the service sends a false response to the interface.
Failed to Edit task	For editing a task the interface sends 2 requests to the service. One for deleting the task from NTScheduler and one for adding it to the Ntscheduler. If for any of these operations the interface receives a false response from the service, this error is generated.
Failed to Delete task	Checkpoint interface sends a request to the service for deleting the task and the service sends a false response to the interface.
Failed to Start Task	Checkpoint interface sends a request to the service for Running the task and the service sends a false response to the interface.
Failed to Stop Task	Checkpoint interface sends a request to the service for stopping the task and the service

If Error Message Is . . .	Then, Possible Cause is . . .
	sends a false response to the interface.
Connection TimeOut	Checkpoint interface does not receive any response from the service within the specified time limit.
Invalid Number of Arguments Provided	The number of arguments for invoking the ChkptUIWrapper.exe is not correct.
Registry Read Failed	Failed to Read a registry key.
Registry Write Failed	Failed to Write to a registry key.
Insert Valid Removable Storage	Removable media such as floppy disk is not present in the drive.
File already exists. Overwrite the existing file?	Occurs in Checkpoint Archive interface when the target file already Exists in the selected path.
File operation Failed	Failed to copy or move file during archive

Checking messages journaled as events

The following table lists messages that are journaled as events in the Event Summary on Station. Some of these messages are the result of detailed errors logged by checkpoint in the Experion error log as noted in the previous *Identifying checkpoint errors logged but not journaled* section.

If Event Message Is . . .	Then, It Means . . .
Save Succeeded	Save succeeded on the tag specified
Restore Succeeded	Restore succeeded on the tag specified
Restore Failed	Restore failed for the tag
Service Started	Checkpoint Service started
Service Stopped	Checkpoint Service stopped
Task Started	Task specified by the tag name started
Task Stopped	Task specified by the tag name stopped
Task Completed	Task specified by the tag name completed

If Event Message Is . . .	Then, It Means . . .
Task Aborted	Task specified by the tag name aborted
Low Disk Space - Older Version Files Deleted	Free space available in the disk is less than the checkpoint preferred free disk Space.
Save completed with Stale data	The save was completed but due to communication problems with some blocks, those blocks had had their prior run time data moved forward into this checkpoint file. Each individual tagged block affected will also be marked **stale**.
Entity Save deferred	The save of the individual node was about to run, but has been deferred since all of the available save thread resources are being used for saves of other nodes still in progress. The save of this node has been queued to be assigned a save thread in the FIFO sequence for any other saves that are also deferred for the same reason. When a save thread becomes available, the first node in the FIFO deferral list will be allocated to that thread, and its save will be started.
Checkpoint Save Resumed For Entity	A node that had its save deferred has now been assigned a save thread in the checkpoint service. The save has been started. This save thread remains allocated to this entity's save until either the save gets completed successfully or fails due to communication or access errors with the entity or the file system. After this, the save thread is returned back to the checkpoint service to determine if it needs to be allocated to the next deferred entity whose save is to be resumed.

Checking warnings and errors Logged but not journaled as events

The following table lists warnings and error messages that are only logged in the Experion error log.

If Warning or Error Message Is . . .	Then, It Means . . .
Periodic Checkpoint Save On Entity SomeName Failed - path not accessible, or insufficient storage space	Checkpoint service is attempting to create an auto-checkpoint file but cannot since there is not enough storage or the path in the Checkpoint share (on the primary server) is not accessible.
Manual Checkpoint Save On Entity *SomeName* Failed - path not accessible, or insufficient storage space	Checkpoint service is attempting to create a manual checkpoint file but cannot since there is not enough storage or the path in the Checkpoint share (on the primary server) is not accessible.
Checkpoint Restore On Entity *SomeName* Failed - path not accessible, or insufficient storage space	Checkpoint Restore interface is attempting to gain access to a checkpoint file that has been selected for restore, but cannot since path to the file is not accessible. Also, this error message can result when the actual checkpoint file can be accessed, but subsequent decompression into the various checkpoint support files cannot result since there is not enough hard drive storage available for the decompression activity.
Checkpoint File *SomeName.cp* failed to be replicated to backup server.	Reported by server file replication when the checkpoint file could not be replicated to secondary server. The problem lies in the server replication activity and not with the checkpoint function.
Checkpoint File *SomeName.cp* failed to be replicated to console station *SomeName*	Reported by server file replication when the checkpoint file could not be replicated to one or more of the console stations. The problem lies in the server replication activity and not with the checkpoint function.
Checkpoint File *SomeName.cp* at path *SomeName* is corrupted.	Checkpoint Restore interface is attempting to use a checkpoint file but determines that contents of file has become physically

If Warning or Error Message Is . . .	Then, It Means . . .
	corrupted (fails CRC32 tests), or the file passes this test, but does not complete the decompression activity due to some corruption in original compression activity.
Checkpoint Restore Not Permitted - File *SomeName.cp* at path *SomeName* is incompatible with target entity's block structure	Checkpoint Restore interface is attempting to use a checkpoint file but determines that the contents of the file are not compatible with the state of the loaded blocks in the control builder monitor. Checkpoint will only permit blocks to be restored that already appear in the Monitoring tree. This prevents ghost blocks from being placed in the node.
QVCS Licensed Found: All periodic schedule tasks reverting to saving only single most recent checkpoint file	Checkpoint service has detected that user has begun using QVCS, and so only saves one version of a checkpoint file for each node that gets subsequently saved.
Checkpoint Restarted Due To Server Failover	Server has failed over and checkpoint service has now restarted on the secondary server machine that has now assumed primary server responsibility.
Manual Checkpoint Save Operation Not Permitted - Restarting Due To Server Failover	Server has failed over and checkpoint service is not yet fully restarted on the secondary server machine that has now assumed primary server responsibility.
Checkpoint Shutting Down	Checkpoint service is in process of being shutdown. This can occur on the primary server at the time that primary server is commanded to swap over to secondary. Furthermore, this can occur if the checkpoint service gets shutdown by a user through the Windows Services panel.
Manual Checkpoint Restore Not Permitted - Entity *SomeName* in invalid state	A node is not permitted to be restored as it is not in a restorable state.
Checkpoint being Aborted - necessary operating files not found in SR	Communication errors with SR, cannot continue the save or the restore.
Checkpoint being Aborted - necessary operating files not found in cache	Memory access problems or cannot find correct files in the node’s checkpoint base subdirectory, cannot continue the save.

If Warning or Error Message Is . . .	Then, It Means . . .
Checkpoint Being Aborted – needed services are no longer running	Checkpoint service has determined that needed services such as CDA, SR, or Windows Task Scheduler is not running. Checkpoint save cannot continue, and/or checkpoint service is shutting down.
Manual Checkpoint Save Not Permitted - Access Denied	Cannot get access to the node targeted by the save operation due to communication problems.
Checkpoint Restore Not Permitted - Access Denied	Cannot get access to the node targeted by the restore operation due to communication problems.
Checkpoint Archiving Not Permitted - Access Denied	Cannot get access to Window's file system components.
Checkpoint file not Found for entity	Checkpoint File missing the specified path

Checkpoint operation with On-Process Migration

Although the On-Process Migration (OPM) Wizard invokes checkpoint rebuild, the checkpoint interface that is provided for OPM usage does not affect any Checkpoint interface functionality available to the user. Checkpoint usage with OPM falls into the following two categories:

On-process server migration

When the server OPM wizard is launched, it will place all autocheckpoint tasks into the **Stopped** state. For any autocheckpoint tasks that are in progress, they will be first aborted and then placed into stopped. The journaled events for such aborted tasks will state that they have been stopped instead of saying completed. This will happen on the existing primary server, just before the secondary server is brought down to be migrated to the new Experion release.

For a R210 to R300.1 migration, the primary server is of release R210, the checkpoint service is stopped on the R210 primary. The user, from this point in time, will not be able to do any checkpoint saves (auto nor manual) on the R210 side. (For future migrations beyond R210/R300.1, when the user proceeds from R300.1 to R3xx, the checkpoint service is not stopped on the older R300.1 side, but all auto-saving is still stopped and prevented. However, the user will still be able to continue to launch **Manual** checkpoint saves on the existing R300.1 primary server.)

After the former secondary server now gets migrated forward into the new Experion release (including migration of its ERDB), the Checkpoint Service runs on that migrated (new release primary) server, but autosaving is still disabled. The checkpoint files on this new release primary (we are still in dual primary mode) must be created in the format understood by checkpoint on this new release before any checkpoint saving can be done on this new release side.

Therefore, the user must manually rebuild all checkpoints of all applicable nodes in this new release primary server. Once this is finished, the user then must perform **Manual** saves (from this new release server side) on each applicable node. This ensures that the user has checkpoint restore coverage in case a controller node fails while the dual primary node is removed and before the former old primary gets installed with the new release and comes up as a secondary. Only after the redundant pair has been fully migrated will auto-checkpoint saving be permitted again by the checkpoint service.

On-process controller migration

Upon launching of the controller OPM wizard, the wizard will require any saving (manual or autocheckpointing) to be completed before it continues. After saves have completed on given controller, the OPM wizard disables any further saves from occurring, and will back up that controller's Checkpoint Base and latest.cp information corresponding to that primary controller as based on the controller's respective Experion release.

OPM Wizard then rebuilds a checkpoint file based on the next Experion release of its firmware or windows .exe (that which the secondary controller is being upgraded to). After secondary controller has been upgraded to the new release code, OPM will use the rebuilt checkpoint (of the new release) and restore that secondary. After the restore, OPM then has the primary controller perform a DSD transfer to the secondary. After that, the secondary controller running on the new release is transitioned to primary.

If at any time the OPM process fails in the above steps, OPM will recover the prior checkpoint base and latest.cp of the former release (that which the primary controller is still running on). Whether OPM fails or passes, it will re-enable the checkpoint saving to be again permitted for that controller.

Upon OPM wizard completion (fail or success), the wizard does not automatically perform a save to refresh the node's latest.cp. The user would need to do this checkpoint save, or have the next autocheckpoint save cycle (applying to this node) take care of getting the node's latest.cp refreshed with run time data.

Fixing common problems

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

Checkpoint file is still marked compatible after change in CAB Type block

If you use one CAB Type block in multiple Control Modules, load all Control Modules to the controller, and do a manual checkpoint save; be aware that making subsequent library changes in the CAB Block Type, and reloading only one of the associated CMs from the **Monitoring** tab leaves the previously saved checkpoint file *compatible* and *restorable*.

However, the monitor form display indication will be different for the CM having the reloaded CAB block as opposed to those CM not being reloaded. For example, in the case where one may add parameters into the CAB Block Type, those parameters will be shown in an defined defaulted condition for the CAB block in the CM that was loaded from Monitoring side. But, for the CMs (having the same CAB block type) that were not reloaded, those CAB monitor forms may show the newly added parameters in an undefined condition.

Note that any reload for a CAB Type change from the **Project** tab makes all checkpoint files *incompatible*.

It is recommended that whenever a CAB block library type gets changed, if one CM having that CAB gets reloaded, then all other CMs having that CAB type should be reloaded.



ATTENTION

This problem may also apply to PHASE Type blocks.

Diagnostic Check	Errors appear on CAB **Monitoring** forms for the associated CMs that were not reloaded after the CAB Type block configuration change.
Cause	• Checkpoint file will continue to match that in the controller. • No errors appear on CAB **Monitoring** form for the associated CM that was reloaded from **Monitoring**.
Solution	• To restore consistency among all associated CMs on the **Monitoring** side, reload the other associated CMs from either the **Project** or **Monitoring** side. • To update run-time information in the checkpoint file, do a manual checkpoint save.

Control Component Display Element Reference

About detail and group displays

The following table lists detail and group display templates that you can use to create your own displays.

Considerations

- You can use the existing display if you configure the name for the given function block to match the name built into the pre-built display template. For example,
  - Name the DEVCTL block in a Control Module DEVCTLA to use the *sysdtldevctla.dsp* detail display template.
  - Name the PID block in a Control Module PIDA to use the *sysdtlpida.dsp* detail display template.
  - Name the DATAACQ block in a Control Module DACA to use the *sysdtldaca.dsp* detail display template.
- For 32-channel Digital Output modules, be sure to change the default group display listed on the configuration form from *sysgrpdx32a.dsp* to *sysgrpdo32a.dsp*.

Block Type	Detail Display	Group Display	Faceplate/ Shortcut Menu
Allen-Bradley Drive Interface - 1305, 1336-PLUS II, GENERIC_DRIVE	sysdtlABDa.htm sysdtlABDb.htm sysdtlABDc.htm sysdtlABDd.htm	sysGrpABDa.dsp	SysDtlABDa_fp.htm
Application Control Environment (ACE)	sysdtlACEA.htm sysdtlACEB.htm	SysGrpACEA.dsp	SysDtlACEA_fp.htm
American Gas Association (AGA) - AGA 3 Orifice Meter, AGA 8 Detail Setup Data	sysdtlaga38detaila.htm sysdtlaga38detailb.htm sysdtlaga38detailc.htm sysdtlaga38detaild.htm sysdtlaga38detaile.htm	sysgrpaga38detail.dsp	sysdtlaga38detaila_fp.htm

Block Type	Detail Display	Group Display	Faceplate/ Shortcut Menu
AGA 3 Orific Meter, AGA 8 Gross Setup Data	sysdtlaga38grossa.htm sysdtlaga38grossb.htm sysdtlaga38grossc.htm sysdtlaga38grossd.htm sysdtlaga38grosse.htm	sysgrpaga38gross.dsp	sysdtlaga38grossa_fp.htm
American Gas Association (AGA) - AGA 7 Turbine Meter, AGA 8 Detail Setup Data	sysdtlaga78detaila.htm sysdtlaga78detailb.htm sysdtlaga78detailc.htm sysdtlaga78detaild.htm sysdtlaga78detaile.htm	sysgrpaga78detail.dsp	Sysdtlaga78detaila_fp.htm
AGA 7 Turbine Meter, AGA 8 Gross Setup Data	sysdtlaga78grossa.htm sysdtlaga78grossb.htm sysdtlaga78grossc.htm sysdtlaga78grossd.htm sysdtlaga78grosse.htm	sysgrpaga78gross.dsp	sysdtlaga78grossa_fp.htm
American Gas Association (AGA) - AGA 9 Ultrasonic Meter, AGA 8 Detail Setup Data	sysdtlaga98detaila.htm sysdtlaga98detailb.htm sysdtlaga98detailc.htm sysdtlaga98detaild.htm sysdtlaga98detaile.htm	sysgrpaga98detail.dsp	sysdtlaga98detaila_fp.htm
AGA 9 Ultrasonic Meter, AGA 8 Gross Setup Data	sysdtlaga98grossa.htm sysdtlaga98grossb.htm sysdtlaga98grossc.htm sysdtlaga98grossd.htm sysdtlaga98grosse.htm	sysgrpaga98gross.dsp	sysdtlaga98grossa_fp.htm
AGA Configuration, Error Code Acronyms	sysdtlagaerrors.htm		
Analog Input Module - 16 Channels	sysdtlai16a.htm sysdtlai16b.htm	sysgrpai16a.dsp	sysdtlai16a_fp.htm
Analog Input Module - 6 Channels	sysdtlaia.htm sysdtlaib.htm	sysgrpaia.dsp	sysdtlaia_fp.htm
Analog Output Module - 8 Channels	sysdtlao8a.htm sysdtlao8b.htm	sysgrpao8a.dsp	sysdtlao8a_fp.htm

Block Type	Detail Display	Group Display	Faceplate/ Shortcut Menu
Analog Output Module - 6 Channels	sysdtlaoa.htm sysdtlaob.htm	sysgrpaoa.dsp	Sysdtlaoa_fp.htm
C300 Controller - Primary	sysdtlc300a.htm sysdtlc300b.htm sysdtlc300c.htm sysdtlc300d.htm sysdtlc300e.htm		sysdtlc300a_fp.htm
C300 Controller - Secondary	sysdtlc300sa.htm sysdtlc300sb.htm sysdtlc300sc.htm sysdtlc300sd.htm		sysdtlc300sa_fp.htm
Generic Control Module	sysdtlcda.htm sysdtlcdb.htm sysdtlcdc.htm	sysDtlCdaGroup.dsp	sysdtlcda_fp.htm
Control Execution Environment	sysdtlceea.htm	sysgrpceea.dsp	Sysdtlceea_fp.htm
Control Execution Environment - Application Control Environment	sysdtlceeacea.htm	sysgrpceeacea.dsp	Sysdtlceeacea_fp.htm
Control Execution Environment - C300	sysdtlceec300a.htm		sysdtlceec300a_fp.htm
Control Execution Environment - Simulation Control Environment	sysdtlceescea.htm	sysgrpceescea.dsp	sysdtlceescea_fp.htm
Control Processor Module - C200 Controller	sysdtlcpma.htm sysdtlcpmastack.htm sysdtlcpmb.htm sysdtlcpmc.htm sysdtlcpmd.htm sysdtlcpmsa.htm	sysgrpcpma.dsp sysgrpcpmsa.dsp	sysdtlcpma_fp.htm sysdtlcpmsa_fp.htm
Series C FIM - Primary	sysdtlCFIMa.htm sysdtlCFIMb.htm sysdtlCFIMc.htm sysdtlCFIMd.htm		sysdtlCFIMa_fp.htm

Block Type	Detail Display	Group Display	Faceplate/ Shortcut Menu
Series C FIM - Secondary	sysdtlCFIMsa.htm sysdtlCFIMsb.htm sysdtlCFIMsc.htm sysdtlCFIMsd.htm		sysdtlCFIMsa_fp.htm
Series C FIM Link	sysdtlCLinka.htm sysdtlCLinkb.htm		sysdtlCLinka_fp.htm
Data Acquisition	sysdtldaca.htm sysdtldacb.htm sysdtldacc.htm	sysgrpdaca.dsp	sysdtldaca_fp.htm
Device Control	sysdtldevctla.htm sysdtldevctlb.htm sysdtldevctlc.htm sysdtldevctld.htm sysdtldevctle.htm sysdtldevctlf.htm sysdtldevctlg.htm	sysgrpdevctla.dsp	sysdtldevctla_fp.htm
Digital Input Module - 32 Channels	sysdtldi32a.htm sysdtldi32b.htm sysdtldi32c.htm	sysgrpdx32a.dsp	sysdtldi32a_fp.htm
Digital Input Module - 16 Channels	sysdtldia.htm sysdtldib.htm sysdtldic.htm	sysgrpdxa.dsp	Sysdtldia_fp.htm
DNET_DEVICE - Generic DeviceNet Device	sysdtldnetdevice.htm	sysgrpdnetdevice.dsp	Sysdtldnetdevice_fp.htm
DNET_IM - DeviceNet Interface Module	sysdtldnetim.htm	sysgrpdnetim.dsp	Sysdtldnetim_fp.htm
Digital Output Module - 32 Channels	sysdtldo32a.htm sysdtldo32b.htm sysdtldo32c.htm	sysgrpdo32a.dsp	sysdtldo32a_fp.htm
Digital Output Module - 8 or 16 Channels	sysdtldoa.htm sysdtldob.htm sysdtldoc.htm	sysgrpdoa.dsp	Sysdtldoa_fp.htm
Digital I/O Module	sysdtldxa.htm sysdtldxb.htm	sysgrpdx32a.dsp sysgrpdxa.dsp	Sysdtldxa_fp.htm

Block Type	Detail Display	Group Display	Faceplate/ Shortcut Menu
Fieldbus Device	sysdtlffdevice.htm	sysgrpffdevice.dsp	Sysdtlffdevice_fp.htm
Fieldbus Device - Analog Input block		sysgrpffai.dsp	
Fieldbus Device - PID block		sysgrpFFPID.dsp	
Fieldbus Interface Module, Secondary	sysdtlFIM.htm sysdtlFIMSA.htm	sysgrpFIM.dsp	SysdtlFIM_fp.htm sysdtlFIMSA_fp.htm
Rail I/O Series A Digital Output Module	sysdtlflexboa.htm sysdtlflexbob.htm	sysgrpdoa.dsp	Sysdtlflexboa_fp.htm
Rail I/O Series A Analog Input Module - 8 Channels	sysdtlflexcai8a.htm sysdtlflexcai8b.htm sysdtlflexcai8c.htm	sysgrpflexai8a.dsp	sysdtlflexcai8a_fp.htm
Rail I/O Series A Analog Output Module - 4 Channels	sysdtlflexcao4a.htm sysdtlflexcao4b.htm sysdtlflexcao4c.htm	sysgrpao4a.dsp	sysdtlflexcao4a_fp.htm
Rail I/O Series A Digital Input Module	sysdtlflexdima.htm sysdtlflexdimb.htm sysdtlflexdimc.htm	sysgrpflexdimcntr.dsp	
Rail I/O Series A Temperature Input Module	sysdtlflextima.htm sysdtlflextimb.htm sysdtlflextimc.htm	sysGrpRailAI8A.dsp	Sysdtlflextima_fp.htm
Fault Tolerant Ethernet Bridge Module - Main, UDP/TCP, IP/ICMP, FTE, ICP Statistics	sysdtlFTEB.htm sysdtlftebb.htm sysdtlftebc.htm sysdtlftebd.htm sysdtlftebe.htm sysdtlftebf.htm	sysgrpFTEB.dsp sysgrpFTEBS.dsp	sysdtlFTEB_fp.htm
HART Analog Input Module	sysdtlhartai8a.htm sysdtlhartai8b.htm	sysgrphartai.dsp	sysdtlhartai8a_fp.htm
HART Analog Output Module	sysdtlhartao8a.htm sysdtlhartao8b.htm	sysgrphartao.dsp	sysdtlhartao8a_fp.htm

Block Type	Detail Display	Group Display	Faceplate/ Shortcut Menu
HART Device - Main, Config Details, Asset Information, HART Data	sysdtlhartdevicea.htm sysdtlhartdeviceb.htm sysdtlhartdevicec.htm sysdtlhartdeviced.htm sysdtlhartdevicee.htm	sysgrphartdevice.dsp	Sysdtlhartdevicea_fp.htm
I/O Link Interface Module, Secondary	sysDtlIOLIMA.htm sysDtlIOLIMSA.htm	sysgrpiolima.dsp	SysDtlIOLIMA_fp.htm sysDtlIOLIMSA_fp.htm
IOLIM - I/O Links	sysDtlIoLink.htm sysDtlIoLinkB.htm	sysgrpiolink.dsp	SysDtlIoLink_fp.htm
I/O Module - Main, Config Details	sysdtlioma.htm sysdtliomb.htm	sysgrpioma.dsp	Sysdtlioma_fp.htm
Fieldbus Interface Link	sysdtlLink.htm	sysgrpLink.dsp	sysdtlLink.htm
Hiway LLPIU Device - Main, Box Hiway Errors, Slot Configuration, Slot Details	sysdtlllpiua.htm sysdtlllpiuap.htm sysdtlllpiub.htm sysdtlllpiuc.htm sysdtlllpiud.htm sysdtlllpiue.htm		sysdtlllpiua_fp.htm
OPC Server	sysdtlopca.htm	sysgrpopca.dsp	
PROFIBUS Interface Module - PBIM_SST	sysDtlPBIM.htm	sysgrppbim.dsp	SysDtlPBIM_fp.htm
SIMATIC Analog Input Module	sysdtlpfbAim.htm	sysgrppfbaim.dsp	SysdtlpfbAim_fp.htm
SIMATIC Analog Output Module	sysdtlpfbAom.htm	sysgrppfbaom.dsp	SysdtlpfbAom_fp.htm
PROFIBUS Interface Module - PBI_DEVICE	sysDtlPfbDevice.htm	sysgrppfbdevice.dsp	SysDtlPfbDevice_fp.htm
SIMATIC Digital Input Module	sysdtlpfbDim.htm	sysgrppfbdim.dsp	SysdtlpfbDim_fp.htm
SIMATIC Digital Output Module	sysdtlpfbDom.htm	sysgrppfbdom.dsp	SysdtlpfbDom_fp.htm

Block Type	Detail Display	Group Display	Faceplate/ Shortcut Menu
PROFIBUS Interface Module - ENCODERDEV	sysDtlPfbEncoder.htm	sysgrppfbencoder.dsp	sysDtlPfbEncoder.htm
PROFIBUS Interface Module - PROFIDRIVEDEV	sysDtlPfbProfiDrive.htm	sysGrpPfbProfiDrive.dsp	SysDtlPfbProfiDrive_fp.htm
Pulse Input Module - Main, Status Data, Configuration	sysDtlPIA.htm sysDtlPIB.htm sysDtlPIC.htm	sysGrpPIA.dsp	SysDtlPIA_fp.htm
PID - Main, Loop Tune, Set Point, PV & OP, Alarms, Connections,	sysdtlpida.htm sysdtlpidb.htm sysdtlpidc.htm sysdtlpidd.htm sysdtlpide.htm sysdtlpidf.htm sysdtlpidg.htm	sysgrppida.dsp	sysdtlpida_fp.htm
PID_PLA - Main, Loop Tune, Advanced, Set Point, PV & OP, Alarms, Connections, Chart	Sysdtlpidpla.htm sysdtlpidplb.htm sysdtlpidplc.htm sysdtlpidpld.htm sysdtlpidple.htm sysdtlpidplf.htm sysdtlpidplg.htm sysdtlpidplh.htm		Sysdtlpidpla_fp.htm
	Sysdtlpidplalta.htm sysdtlpidplaltb.htm sysdtlpidplaltc.htm sysdtlpidplaltd.htm sysdtlpidplalte.htm sysdtlpidplaltf.htm sysdtlpidplaltg.htm sysdtlpidplalth.htm		Sysdtlpidplalta_fp.htm
HART PM I/O Module	sysdtlPmioIOC.htm		sysdtlPmioIOC_fp.htm
PM I/O Processor - A, B	sysdtlPmioIOP.htm sysdtlPmioIOPB.htm		sysdtlPmioIOP_fp.htm
POSPROP			sysdtlpospa_fp.htm

Block Type	Detail Display	Group Display	Faceplate/ Shortcut Menu
Ramp Soak Profiles - 1 to 10	sysdtlprofile1.htm sysdtlprofile2.htm sysdtlprofile3.htm sysdtlprofile4.htm sysdtlprofile5.htm sysdtlprofile6.htm sysdtlprofile7.htm sysdtlprofile8.htm sysdtlprofile9.htm sysdtlprofile10.htm		
Rail I/O Series H Digital Input Module - 16 Channels; Main, Status Data, Configuration	sysDtlRailBI16A.htm sysDtlRailBI16B.htm sysDtlRailBI16C.htm		sysDtlRailBI16A_fp.htm
Rail I/O Series H Digital Output Module - 4 Channels; Main, Status Data, Configuration	sysDtlRailBO4A.htm sysDtlRailBO4B.htm sysDtlRailBO4C.htm		sysDtlRailBO4A_fp.htm
Rail I/O Series H Analog Input Module - 8 Channels; Main, Status Data, Configuration	sysDtlRailCAI8A.htm sysDtlRailCAI8B.htm sysDtlRailCAI8C.htm	sysGrpRailAI8A.dsp	sysDtlRailCAI8A_fp.htm
Rail I/O Series H Analog Output Module - 8 Channels; Main, Status Data, Configuration	sysDtlRailCAO8A.htm sysDtlRailCAO8B.htm sysDtlRailCAO8C.htm		sysDtlRailCAO8A_fp.htm
Rail I/O Series H Temperature Input Module - 8 Channels; Main, Status Data, Configuration	sysDtlRailTAI8A.htm sysDtlRailTAI8B.htm sysDtlRailTAI8C.htm		sysDtlRailTAI8A_fp.htm

Block Type	Detail Display	Group Display	Faceplate/ Shortcut Menu
RAMPSOAK - Main, Profile Graph, Profile Data, Alarms, Connections	sysdtlrampa.htm sysdtlrampb.htm sysdtlrampc.htm sysdtlrampd.htm sysdtlrampe.htm sysdtlrampf.htm	sysgrprampa.dsp sysgrprampb.dsp	sysdtlrampa_fp.htm
Redundancy Module - Main, RM Profile, Configuration, Synch, Chassis Profile	sysdtlrma.htm sysdtlrmb.htm sysdtlrmc.htm sysdtlrmd.htm sysdtlrme.htm	sysgrprma.dsp	Sysdtlrma_fp.htm
Simulation Control Environment (SCE)	sysdtlscea.htm	sysgrpscea.dsp	Sysdtlscea_fp.htm
Series C I/O	sysdtlSCIOMa.htm sysdtlSCIOMb.htm sysdtlSCIOMc.htm		sysdtlSCIOMa_fp.htm
Sequential Control Module (SCM) - Main, State Diagram, Recipe 1-16, Recipe 17-33, Recipe 34-50, History 1-16, History 17-33, History 34-50, Handler	sysdtlscma.htm sysdtlscmb.htm sysdtlscmc.htm sysdtlscmc2.htm sysdtlscmc3.htm sysdtlscmd.htm sysdtlscmd2.htm sysdtlscmd3.htm sysdtlscme.htm sysdtlscmf.htm sysdtlscmg.htm	sysgrpscma.dsp	sysdtlscma_fp.htm
Serial Interface - 32 Channels; Main, Channels	sysdtlsia.htm sysdtlsib.htm	sysgrpsia.dsp	Sysdtlsia_fp.htm
Simulation I/O Link Interface Module - SIMIOLIM	sysDtlIOLIMA.htm	sysgrpSimiolim.dsp	SysDtlIOLIMA_fp.htm
Simulation I/O Link - SIMIOLINK	sysDtlSimioLink.htm	sysgrpSimiolink.dsp	SysDtlSimioLink_fp.htm

Block Type	Detail Display	Group Display	Faceplate/ Shortcut Menu
PROFIBUS Interface Module - SIMOCODE3UF5	sysDtlSIMOCODE3UF5.htm	sysGrpSIMOCODE3UF5.dsp	sysDtlSIMOCODE3UF5_fp.htm
PID-PL - Main, Looptune, Advanced, Set Point, PV & OP, Alarms, Connections, Chart	sysdtlpidplmain.dsp sysdtlpidpllooptune.dsp sysdtlpidplpvop.dsp sysdtlpidplsetpoint.dsp sysdtlpidpladvanced.dsp sysdtlpidplalarms.dsp sysdtlpidplchart.dsp sysdtlpidplconnections.dsp	sysgrppidpl.dsp	sysdtlpidplmain_fp.htm
PM I/O Processors		sysGrpPmioAO8.dsp sysGrpPmioAO16.dsp sysGrpPmioAO16HART.dsp sysGrpPmioDI24V.dsp sysGrpPmioDI.dsp sysGrpPmioDISOE.dsp sysGrpPmioDO16.dsp sysGrpPmioDO32.dsp sysGrpPmioHLAI.dsp sysGrpPmioHLAIHART.dsp sysGrpPmioLLAI.dsp sysGrpPmioLLMUX.dsp sysGrpPmioRHMUX.dsp sysGrpPmioSTIMV.dsp	
PM HART I/O Channels		sysGrpPmioHAICHANNEL.dsp sysGrpPmioHAOCHANNEL.dsp	

Honeywell

Honeywell International
Process Solutions
2500 West Union Hills
Phoenix, AZ 85027