Honeywell

Experion PKS
Application Control Environment User's Guide

EPDOC-XXX4-en-431A
February 2015

Release 431

Honeywell

Document	Release	Issue	Date
EPDOC-XXX4-en-431A	431	0	February 2015

Disclaimer

Contents

1 About This Document

This document describes the Application Control Environment (ACE) node and its configuration, installation, and operation. The document includes the basic information for the interface of the ACE to the OPC Gateway node and the CAB function

Revision history

Revision	Date	Description
A	February 2015	Initial release of document.

2 Getting Started

This section identifies the tasks you need to complete to add an Application Control Environment (ACE) controller node to your Experion system. If you have previously installed an ACE controller node, you may choose to skip those tasks that describe ACE and its functionality and proceed to the actual installation procedures. Be aware that Experion release R210 includes new ACE controller functionality. The following table lists the tasks in the order in which they are to be completed and provides links to applicable documents where appropriate.

Task	Link or Reference
Review the structure of the ACE node and its functionality	See the listings under "ACE Topics" on page 10
Install and set up the ACE Node computer	See ACE computer manufacturer's documentation.
Connect the ACE node to the Experion network	"Connecting ACE to the Experion Network" on page 17
Install the Application Control Environment (ACE) software version R210 or greater.	See the *Software Installation and Upgrade Guide*
Configure and load the ACE node and the ACE/CEE	"Creating an ACE and CEE" on page 52

Related topics

"ACE Topics" on page 10

2.1 ACE Topics

Topic	Link or Reference
Overview	“Overview” on page 12
System configuration considerations	“Configuration considerations” on page 16
Software licensing	“ACE Licensing” on page 18
Memory and processing resources	“CEE/ACE Memory and Processing Resources” on page 28
FB scheduling and execution	“Scheduling and Execution” on page 19
Process special	“Blocks with No Periods: Process Specials” on page 25
Custom Algorithm Blocks (CAB)	“Configuring ACE for CAB” on page 30
Peer-to-peer functionality	“Peer-to-Peer Considerations” on page 31
LCN Prefetch for ACE CMs	“About LCN prefetch” on page 36
Control Functions/Insertion Points	“Insertion Point Support” on page 68
Operations	“Interacting with ACE” on page 70
Troubleshooting and Maintenance	“Recovering from ACE node power failure” on page 80

3 Application Control Environment (ACE)

Related topics

“Overview” on page 12

3.1 Overview

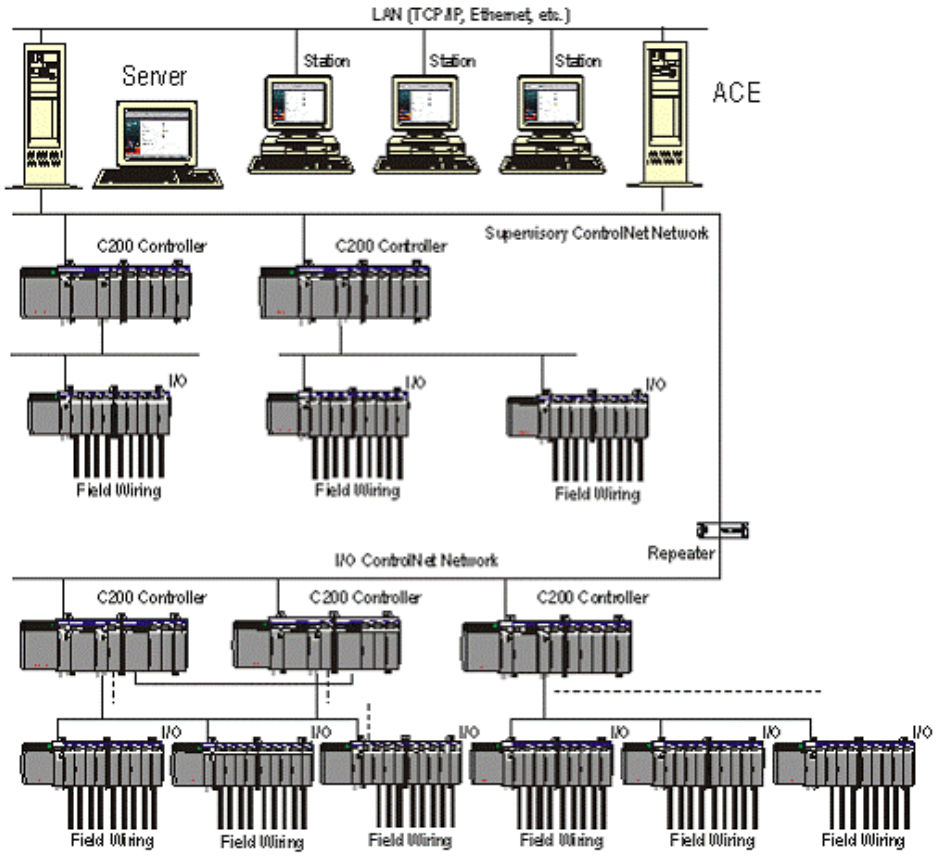
The ACE controller mirrors the basic operations of a Control Processor Module (CPM). It provides the additional capability of communicating with OPC Servers through a Fault Tolerant Ethernet (FTE) or redundant or non-redundant Ethernet network. The ACE program runs on a server grade personal computer using a Windows 2003 Server operating system. Users can optionally connect an ACE controller directly to a supervisory Fault Tolerant Ethernet (FTE) or ControlNet network to support peer-to-peer communications with C200 Process Controllers. Or, to the FTE network for C300 Controllers.

ACE can be included in the following network topologies:

- As part of an existing Total Plant Solutions (TPS) system with Application (APP) node, Network Interface Module/Universal Control Network (NIM/UCN) and/or Hiway Gateway (HG)/TDC2000 Hiway (See “Interface between ACE and TPS” on page 13)
- As part of a new Experion system with C200 Controllers and FTE Bridge
- As part of a new Experion system with C300 Controllers and FTE network
- As part of an existing Experion system with C200 Controllers and/or C300 Controllers

The location of the ACE in a typical medium-size control system with ControlNet communications is illustrated below:

Table 1: ACE as Part of an Experion Control System



3.1.1 ACE and OPC Gateway

The ACE controller supports OPC Server communications through the OPC gateway. See *Using OPC Gateway to Interface ACE to an OPC Server* for more information.

3.1.2 ACE process architecture

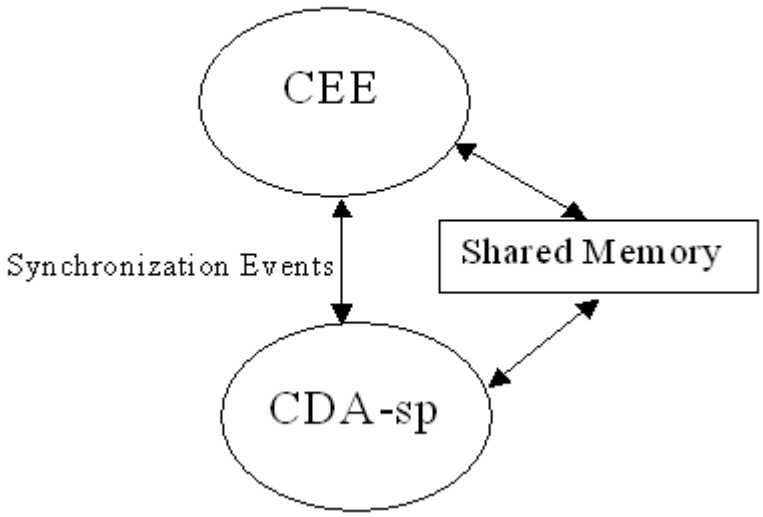
The figure below illustrates the process architecture of the ACE controller. It includes the following subsystems:

- Control Execution Environment (CEE)
- Control Data Access - supervisory (CDA-sp)

The subsystems communicate through two basic mechanisms:

- Shared memory
- Windows Synchronization Events

Table 2: ACE Process Architecture



3.1.3 ACE/CEE database

The ACE/CEE architecture does not contain any permanent, hard-disk-based database. You load ACE/CEE configuration information and command its state (CEE state of Run) through Control Builder.

3.1.4 Interface between ACE and TPS

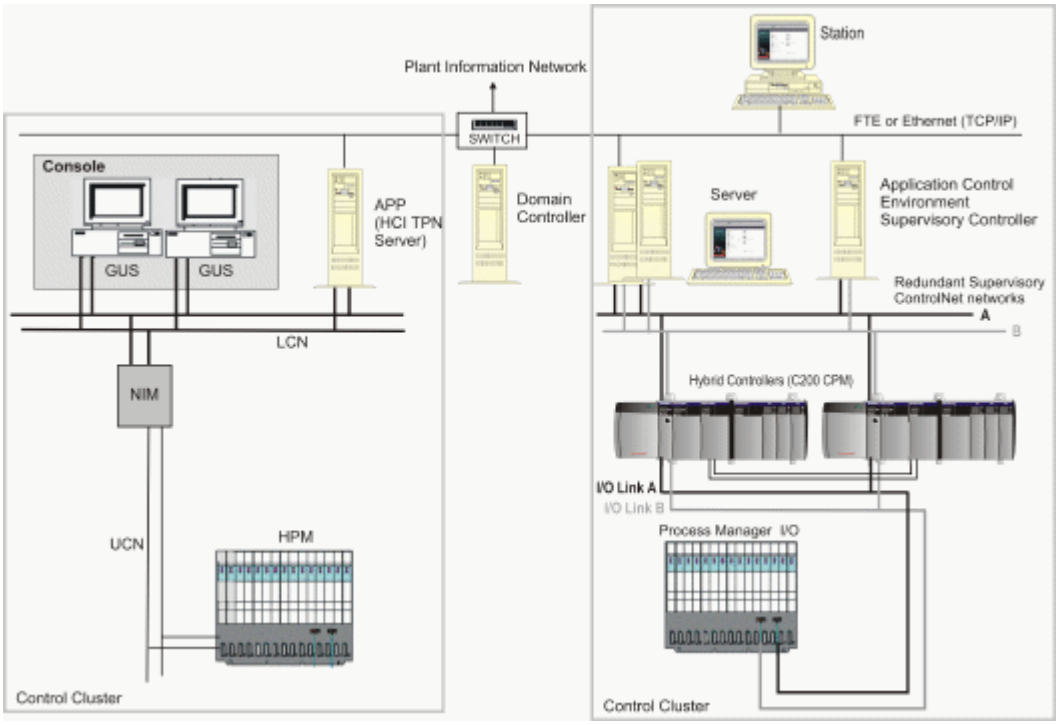
The ACE controller interfaces to the TPS system through an Application (APP) node running Total Plant Network (TPN) server as an OPC server. The TPN server supports both OPC data access and Honeywell Communications Infrastructure (HCI) extensions to OPC. The ACE controller recognizes when it is communicating with an OPC server that includes HCI, such as the APP node, and provides appropriate functional support.

Tip

The additional HCI related functions are transparent to the Control Modules and Sequential Control Modules in control strategies that are executing in the ACE controller. No changes are required in control strategy configurations through Control Builder whether the OPC server being accessed supports HCI or not.

The relation between ACE and TPS is illustrated below.

Table 3: Relationship between ACE and TPS



4 Planning and Design

Related topics

4.1 Configuration considerations

Points to consider when implementing ACE controllers.

Refer to the Experion ACE Specification (EP03-310-310) for current performance specification ratings.

- The ACE controller requires system Server and Station programs to support Operator Interface, History, and other functions, just like the Control Processor Module (CPM).
- Only one ACE environment is supported per dedicated computer running Windows 2003 Server operating system.
- The ACE is a **soft** controller platform that is subject to the **restrictions** of the host operating system.
- Do **not** load system Server/Client, Station, or Control Builder program on a dedicated ACE controller computer.
- Up to 5 ACEs are supported per Experion server connected through FTE (in addition to the C200 and C300 controllers supported by Experion server).
- Redundant ACE controllers are **not** supported.
- The ACE controller can peer-to-peer with other ACE controllers connected to the same Server over Fault Tolerant Ethernet or Ethernet network.
- Beginning in Experion R311, ACE supports a prefetch function for OPC gateways talking to a Local Control Network (LCN) to give users the ability to request peer data just before the data is required by a slow executing Control Module (CM) in ACE CEE.
- Starting with Experion R430, peer-to-peer communication between ACE Controller and EHPM Controller is supported. To establish peer-to-peer communication between ACE Controller and EHPM Controller, refer to the *Control Building User's Guide*.
- The ACE supervisory controller that has a direct connection to the ControlNet or FTE supervisory network can peer-to-peer with C200 controllers that belong to the same Server. The same is true for C300 controllers on the FTE supervisory network.
- ACE is **not** qualified to run on the Experion server node.
- Up to 2 ACEs are supported directly connected to the same ControlNet.
- A single ACE can have a maximum of 30 connections to components connected via FTE Bridge or via PlantScape ControlNet. Connected components include C200s, and FIMs. This includes C300s and Series C FIMs connected through a FTE network.
- Also see “Configuring ACE for CAB” on page 30.

4.2 Connecting ACE to the Experion Network

4.2.1 ACE LAN connection types

ACE is connected to the Experion LAN using

- FTE
- Experion Redundant Ethernet
- Experion Non-redundant Ethernet

4.2.2 ACE supervisory network connection types

ControlNet connection

ACE requires the installation of RSLinx software and a PCIC card for ControlNet communication with C200 through a ControlNet Interface Module.

FTE connection

ACE requires the installation of FTE software and a NIC card for FTE communication with C200 through an FTE Bridge Module or C300 through the FTE network.

4.2.3 Restrictions

Attention
The following restrictions apply when configuring an Experion system using the ACE

- The ACE node is **not** supported when connected to an Ethernet supervisory network configuration.
- FTE connected Servers do **not** support Ethernet supervisory network, since FTE does **not** run on nodes with three network interface card ports.
- The ACE node can be directly connected to the supervisory ControlNet or Fault Tolerant Ethernet segment, if required to support peer-to-peer communication with C200 Controllers or FTE only for C300 Controllers.
- All ACE to Server traffic uses the Ethernet or FTE media link. Only ACE peer-to-peer connections with C200 Controllers will use the ControlNet or FTE supervisory network link. The C300 Controllers use FTE network only.
- Depending on the application, increased memory consumption may result in complete ACE failure due to hardware memory limitations and underlying operating system dynamics. Please refer to the Custom Algorithm Block and Custom Data Block User's Guide for more information about *Organizing CAB programs for best performance*.

4.2.4 Connecting to the network

- Refer to the *Fault Tolerant Ethernet Bridge Implementation Guide* for more information about FTE network connections.
- Refer to the *ControlNet Installation Guide* for more information on ControlNet connections.
- Refer to the *Fault Tolerant Ethernet Overview and Implementation Guide* for more information about setting up FTE networks.

4.3 ACE Licensing

Related topics

“Licensing check” on page 18

4.3.1 Licensing check

Application Controller Environment Software and Function Block Libraries are subject to licensing. The license always applies to the Experion server to which the ACE Server is connected. Licenses are not transferred between servers. There are two types of licenses, as indicated per model number:

A - a fixed number of instances can be loaded to the system, regardless of controller assignment,

B - the Function Block library or feature is enabled; the number of instances of a Function Block does not matter.

Use the following table to verify the ACE software license:

Model Number	Model Description	License Type
TC-SWSC01[1]	ACE Base Software License	A
TC-OPCL01[2]	OPC Extension Library	B
TC-UCNL01[2]	UCN Output Extension Library	B

(1) The ACE Base software requires a dedicated Windows 2000 Server PC

(2) TC-OPCL01 and TC-UCNL01 are additional libraries to the ACE Node. These require TC- SWSC01

4.4 Scheduling and Execution

Related topics

4.4.1 Scheduling and execution topics

Topic	Description	For Details:
Period	The amount of time between two consecutive executions of a CM or SCM block.	“Period” on page 19
Phase	Determines the set of cycles in which a container block executes.	“Phase” on page 20
Slow execution periods	ACE functionality which allows extension of the base periods by executing container blocks at specified hours and minutes from the startup of the ACE node	“Slow execution periods” on page 20
Execution Order	The time sequence in which container blocks execute	“Execution order” on page 21
Overruns	Failure of container blocks to execute within the specified time cycle	“Overruns” on page 22
Execution Balancing	Procedure by which overruns are eliminated or minimized by automatically assigning scheduling parameters to container blocks.	“Execution balancing” on page 23
Process Special	Configuration option that permits an Operator, CAL program, CM, or SCM to trigger the execution of another Control Module.	“Blocks with No Periods: Process Specials” on page 25

4.4.2 Period

The period of a CM or SCM container block is the amount of time between two consecutive executions of the block.

The periods available to container blocks in the current Experion CEEs are summarized below:

5 ms CEE/CPM	50 ms CEE/CPM	500 ms CEE/ACE
5, 10, 20, 50, 100 and 200 ms (Default is 200 ms)	50, 100, 200, 500, 1000, and 2000 ms (Default is 1000 ms)	500 ms, 1s, 2s, 5s, 10s, 20s, 30s, *1min, 2min, 5min, 10min, 20min, 30min, 1hr, 2hr, 4hr, 8hr, 12hr, 24hr.* (Default is 2s; *italicized values require use of phasing parameters as described below*)

When a CM or an SCM is assigned to any CEE, the selected period is validated against the permitted execution periods for that CEE and that module. The assignment is disallowed if the period is not supported. For example, any attempt to assign a CM executing at a period greater than 2 seconds to a CPM's CEE is rejected.

If a CM is already assigned to a C200 and the user attempts to configure its period to one that is only supported by ACE, the change is also rejected. The same is true if the CM has already been loaded to a C200 or C300 and the change is made on-line.

Since SCMs do not support periods greater than 20 seconds, any attempt to change the period to a slower period is rejected. This includes changes at configuration time, assignment time and load time.

4.4.3 Phase

The phase parameter determines the set of cycles in which a container block executes.

ACE has a base cycle of 500 ms and in each phase blocks are executed and then parameter requests originating from peer environments and operator displays are serviced. CMs are assigned a phase in which their execution starts. From that point on, the CMs are executed at a frequency based on their configured period. Currently, ACE supports 120 phases, allowing for periods up to 1 minute (120 x 500 ms) using the PHASE parameter alone. Longer periods require the use of the PHASEMIN and PHASEHOUR parameters as noted in the slow execution periods discussion below.

4.4.4 Slow execution periods

To support slower periods, two new phase parameters are added to allow the user to select which minute of an hour to begin execution and which hour in a 24-hour time-span to begin execution. This is in addition to the selection of one of the 120 phases. The new parameters, PHASEMIN and PHASEHOUR are not valid for every possible period. If the parameter is not valid, it will remain at its default of -1. The two tables below show all the periods with the valid ranges and values for each of the PHASE related parameters.

Table 4: Period and Phase Ranges

Period	PHASE Range	PHASEMIN Range	PHASEHOUR Range
500 ms, 1s, 2 s, 5 s, 10 s, 20 s, 30 s	0-119	N/A	N/A
1 m, 2 m, 5 m, 10 m, 20 m, 30 m, 1hr	0-119	0-59	N/A
2 hr, 4hr, 8 hr, 12 hr, 24 hr	0-119	0-59	0-23

Table 5: Period and Valid Phase Assignment

Period	PHASE	PHASEMIN	PHASEHOUR
500 ms	0	N/A	N/A
1s	0,1	N/A	N/A
2 s	0 thru 3	N/A	N/A
5 s	0 thru 9	N/A	N/A
10 s	0 thru 19	N/A	N/A
20 s	0 thru 39	N/A	N/A
30 s	0 thru 59	N/A	N/A
1m	0 thru 119	0	N/A
2 m	0 thru 119	0,1	N/A
5 m	0 thru 119	0-4	N/A

Period	PHASE	PHASEMIN	PHASEHOUR
10 m	0 thru 119	0-9	N/A
20 m	0 thru 119	0-19	N/A
30 m	0 thru 119	0-29	N/A
1 h	0 thru 119	0-59	0
2 h	0 thru 119	0-59	0,1
4 h	0 thru 119	0-59	0-3
8 h	0 thru 119	0-59	0-7
12 h	0 thru 119	0-59	0-11
24 h	0 thru 119	0-59	0-23

4.4.5 Execution order

Control Modules that are executed within the same phase are executed in the order defined by the parameter, ORDERINCEE. If two CMs are created and CM1 must execute before CM2, they both could be configured to begin execution in phase 0 and CM1 would have an ORDERINCEE value less than that of CM2. Now if there was a CM3 that needed to execute every 20 seconds, starting in phase 0, its ORDERINCEE would also be considered when these three CMs lined up in the same phase, which in this case would be phase 0. If CM3 is not dependent on any other CM executing before it or after it, then the value for ORDERINCEE could be left at its default.

With the addition of slower periods and 120 phases, nothing changes with respect to ORDERINCEE. If two CMs, CM4 and CM5, must execute once a minute with CM4 executing first, then both can be configured to start in phase 0 with CM4 having the lower value for ORDERINCEE. When CM4 and CM5 execute in the same 500-ms time slice as CM1, ACE will use the ORDERINCEE to determine which one to execute first. No priority is given to the Control Module with the faster period.

Tip

If several CMs must execute once a minute their execution could be ordered by selecting different phases as appropriate. This would also help balance the execution load over the minute if “Execution balancing” on page 23 were not selected.

4.4.6 Scheduling example

The scheduling execution will be described using the following configuration:

Table 6: Configuration Example

CM NAME	PERIOD	ORDERINCEE	PHASE	PHASEMIN	PHASEHOUR
CMSEC1	1 sec	5	1	N/A	N/A
CMSEC2	/sec	30	0	N/A	N/A
CMSEC3	30 sec	50	0	N/A	N/A
CMMIN1	20 min	20	119	5	N/A
CMMIN2	1 min	15	1	0	N/A
CMHR1	8 hour	10	119	2	5
CMHR2	24 hour	40	1	5	5

The following table is used to illustrate the execution schedule resulting from the above configuration. All CMs are activated at the same time the CEE is activated. This time is considered hour 0, minute 0.

Table 7: Execution Order

Phase	0 (500 ms)	1 (500 ms)	2 (500 ms)	3 (500 ms)	..	29 (500 ms)	..	117 (500 ms)	118 (500 ms)	119 (500 ms)
Order	CMSEC2	CMSEC1	CMSEC2	CMSEC1		CMSEC1		CMSEC1	CMSEC2	CMSEC1
	CMSEC3	CMMIN2 (1)		CMSEC2		CMSEC2		CMSEC2		CMHR1 (3)
		CMSEC2				CMSEC3				CMMIN1 (4)
		CMHR2 (2)								CMSEC3

1. Execution begins on hour 0, minute 0 and occurs every minute after that
2. Execution begins on hour 5, minute 5 and every 24 hours after that
3. Execution begins on hour 5, minute 2 and every 8 hours after that
4. Execution begins on hour 0, minute 2 and occurs every 20 minutes after that

4.4.7 Overruns

An overrun occurs when any container blocks assigned to a cycle fail to execute within the cycle. Overruns can generate alarm events as noted below.

ACE overrun alarms are reported when two consecutive 60-second macro-cycles have at least one cycle with an overrun. *The alarm is cleared when two consecutive 60-second macro-cycles have no cycle with an overrun*

In addition to the events, statistics for 120 base cycles are kept to indicate the number of overruns in the current hour and previous hour. This information is shown in a CPU Overrun tab that appears in Control Builder as well as the Station Detail Display for a CEE block. This display also shows the following additional statistics:

- Number of overruns per cycle in the last 24-hour cycle
- Number of overruns per cycle in the previous 24-hour cycle
- Max Overruns per cycle in any given 24-hour period

All of these statistics are calculated since CEE activation time or since the last time the user reset the statistics via a command.

There is also a CPU Loading tab that shows the average CPU usage per cycle as well as the max CPU usage per cycle. This tab appears on the CEE monitoring form as well as the Station Detail Display. The tab shows the CPU utilization as a percent of the 500 ms cycle time and the maximum CPU usage for 120 cycles of CPU usage.

Be aware that pushing data to a CAB block using Value Custom Data Parameters (CDP) could cause an increase of memory usage on the ACE, which may result in an overrun when the system tries to reclaim this memory. This is especially true in cases where the CAB block is placed in a DORMANT state and it no longer processes data being pushed to it These Custom Data Parameters are defined as String or Time Data Types.

If you encounter memory increases related to running CAB blocks, you can limit memory usage by using Parameter References instead of Value Custom Data Parameters. Using Parameter References allow you to control when data is being fetched and prevents the increase in memory in those conditions where the CAB block is DORMANT.

4.4.8 Analysis of overrun data

The CPU usage and overrun displays are used to diagnose problems as explained with the following example.

If an ACE has 1000 CMs to execute, 750 configured to run every second, 250 configured to run every minute in phase 0 then once a minute the worst-case CPU usage is realized. Let's say that the 750 most frequently executing CMs can execute fine but every minute the additional 250 CMs cause the CPU utilization for cycle 0 to go over 500ms. This situation will not result in an alarm. However, the number of overruns per current hour displayed in row 0 will increment once a minute until the problem is fixed. The next step is to find all the Control Modules that execute in the cycle with the overrun indication.

For each CM and SCM a read only arrayed parameter of Booleans is supported to indicate in which cycles (0-119) the module will execute. This parameter is calculated each time it is requested. For the CMs in the previous example that execute once a second in phase 0, every other array element starting with 0 will return a TRUE value. For the CMS that execute every minute, only array element zero returns TRUE. This parameter allows the user to find all CMs that execute in the cycle or cycles that overran.

Once the CMs that execute in the cycles with overruns are identified, the user must re-distribute their execution. If the execution balancing option was not used initially, it is recommended that all CMs without order dependency be loaded using the execution balancing option described below If execution balancing is not desired, then the user must re-distribute the CMs and SCMs by phasing them (e.g. change Phase parameter to achieve a more balanced execution).

4.4.9 Execution balancing

CEE provides execution balancing for distributing the execution of CMs and SCMs over the 60-second macro-cycle (120 cycles), a 1-hour macro-cycle and the 24-hour macro-cycle based on the total number of modules to execute in each of 500-ms cycles in a minute as well as the number of modules to execute in each minute of an hour and each hour of the day. Execution balancing is not done based on CPU resource requirements.

For all periods, the user can request execution balancing over the 120 cycles, by setting the PHASE parameter to -1 on the CM or SCM configuration form and then doing a load. When the CEE receives the configuration, it chooses a PHASE, changes the value from -1 to the chosen phase and places the CM or SCM on the execution chain according to the specified ORDERINCEE. For the slower periods, the user can request execution balancing over a 24-hour period by setting PHASEHOUR to -1. Execution can also be balanced over a given hour by selecting PHASEMIN of -1. Note that the default value for all of these parameters is -1.

When CMs/SCMs are loaded, the module count for each 500-ms cycle, each minute and each hour are incremented. When CMs/SCMs are deleted, the appropriate counts are decremented.

A checkpoint restore or a reload from project or monitoring results in the same execution schedule as long as both of the following actions are taken after a load with execution balancing requested or after any on-line changes to Phase/Period parameters:

- Upload and update to project - updates monitor and project sides of database with latest scheduling information.
- Checkpoint save - updates checkpoint file with latest scheduling information.

Attention

If two CMs or SCMs execute at the same frequency and have an execution dependency then their execution order in relationship to one another cannot be guaranteed if execution balancing is selected. The ORDERINCEE would be used to order the execution in relationship to other CMs/SCMs executing in the same cycle but there is no guarantee that the two CMs or SCMs will be placed in the same phase. For this case, the user must configure the PHASE and ORDERINCEE parameter. To get the best execution balancing based on counts, any CMs with order dependencies should be loaded first so the others can be balanced around them.

Using the “Scheduling example” on page 21, with the same ORDERINCEE but with PHASE, PHASEMIN and PHASEHOUR all set to -1, the schedule might end up as follows assuming that the CMs are loaded in the order they appear in the table:

Table 8: Execution Order

Phase	0	1	2	3	4	5	6	7	8	9
Order	CMSEC 1	CMSEC 2	CMSEC 1	CMMIN 1 (1)	CMSEC 1	CMMIN2 (2)	CMSEC 1	CMHR1 (3)	CMSEC 1	CMSEC 2
	CMSEC 2	CMSEC 3	CMSEC 2	CMSEC 2	CMSEC 2	CMSEC2	CMSEC 2	CMSEC 2	CMSEC 2	CMHR2 (4)

1. Execution begins on hour 0, minute 0 and occurs every 20 minutes after that
2. Execution begins on hour 0, minute 1 and every minute after that
3. Execution begins on hour 1, minute 2 and every 8 hours after that
4. Execution begins on hour 2, minute 3 and occurs every 24 hours after that

4.5 Blocks with No Periods: Process Specials

The ACE provides a configuration option and a scheduling mechanism for an Operator, CAB program, CM or SCM to trigger another Control Module to execute.

ACE supports an execution period value of NOPERIOD to support on-demand CM execution for CMs that do not need to execute periodically. When a CM with this option is loaded to ACE, it is not assigned to any cycle, its PHASE parameter is defaulted to 0 and the ORDERINCEE is not applicable. The Control Module must be activated just like any other Control Module in order to execute. Execution however, only occurs when an operator, program or another function block does a store to a Process Special Trigger parameter. After the CM executes, the parameter is reset and processing can be triggered again. Multiple stores done prior to the CM's execution are not queued. Storing a FALSE or zero value to this parameter when it is TRUE will cause the Process Special request to be cancelled. An additional process special parameter is supported which requires program access level. This allows SCMs and other function blocks, such as CAB to trigger execution to occur after x seconds where x is the integer value stored to the parameter. Similar to the Boolean parameter, subsequent stores to this parameter are ignored until the block executes. Storing a value of zero to this parameter causes it to execute immediately. If a process special has been triggered using either of the trigger parameters, all stores to either of the parameters are rejected until the block executes.

Attention

Process Special is not supported for SCMs or for any CMs loaded to a C200. For SCMs, the selection of NOPERIOD is rejected at configuration time. For CMs, the selection of NOPERIOD is rejected at assign time and load time if the CM is assigned to a C200. On-line changes to a value of NOPERIOD are also rejected for SCMs and any CM running in a C200.

4.5.1 Initiating a process special request

A process special request is triggered using either the BPS or the BPSDELAY parameters on a CM.

BPS: Storing a TRUE value to the BPS parameter initiates a process special request. Consecutive stores of a value TRUE to the BPS parameter once a process special request is in progress are ignored. However, as soon as a process special request is complete, the next store of a TRUE value to BPS initiates a new process special request. Storing a FALSE value to the BPS parameter cancels any outstanding process special request on a CM.

BPSDELAY: Storing a non-negative value to the BPSDELAY parameter is used to initiate a timed delay process special request. The process special request is initiated on the write to the parameter but it isn't executed until after a delay equal to the number of seconds written to the BPSDELAY parameter. While the BPSDELAY parameter is a floating-point value, the actual delay time is adjusted to be in increments of the CEE base cycle. Writing the value zero to the BPSDELAY parameter is equivalent to writing a TRUE value to the BPS parameter. As with the BPS parameter, additional writes to the BPSDELAY parameter are ignored once a process special request is initiated. There is no way to cancel a process special request using the BPSDELAY parameter. However, a process special request initiated with the BPSDELAY parameter can be canceled by writing a FALSE to the BPS parameter.

The process special execution of CM blocks is carried out based on the order in which the process special requests are initiated and taking into account delays specified in the request in the following fashion.

- A process special request initiated by BPS is executed after any CM already scheduled for process special by BPS and after any CM scheduled for process special by BPSDELAY where the delay time remaining is equal to zero.
- A process special request initiated by BPSDELAY is executed after any CM with a remaining delay less than or equal to the request delay time and then in the order in which process special requests were initiated.

When a process special request is made, the CM is executed on the first available CEE cycle, based on the above description, after any timed delay has passed. An available CEE cycle is determined based on time remaining in a CEE base cycle after all scheduled blocks have been executed. In addition, only a limited number of process special CMs will be executed in any one CEE base cycle regardless of time remaining in the cycle. This is to avoid overloading CDA with a large number of request-response peer data requests.

No effort will be made to provide between block execution "store retries" when a block is executed by process special. This means that if a store by inside connector fails, there will be no way to retry the store until the next time the block executes, no matter how much later the next execution occurs.

When a CM has a period of NONE and it is being executed by process special, it's up to the control engineer to scale period-related values in blocks placed on the CM. The control engineer does this by writing the SCALEPERIOD parameter on the CM with the desired scaling period. The default scaling period for a CM with a period of NONE is equal to one minute unless another value is set using the SCALEPERIOD parameter on the CM.

You can get a Control Module with an execution period of None to process itself, after being awakened from process special parameters, by first storing a FALSE or zero value to the BPS parameter to cancel the process special request in this Control Module

4.5.2 Process special and periodic CMs

A process special request is normally associated with a CM having a period of NONE. However, a request can be initiated on a CM that has a period other than NONE. If a process special request is made on a CM that has a period other than NONE, one of two things will happen.

- If the CM process special execution occurs before the next scheduled periodic execution of the block, then the CM will perform both the process special and the periodic block executions.
- If the next scheduled periodic execution of the CM occurs before the process special execution takes place, because of a process special time delay or a simple race condition, then occurrence of the periodic execution will cause the process special execution to be cancelled.

4.5.3 Process special control loop considerations

If you associate a process special with a Control Module that contains a control loop, which includes a DATAACQ block connected to a PID block, be sure you configure the ORDERINCM parameter value for the DATAACQ block to be **less than** the ORDERINCM value set for the PID block. This allows the DATAACQ block to get the latest input value for the cycle before the PID block fetches its PV data.

4.5.4 Process special for slow periods

Typically, process special is used to trigger execution of blocks that do not need to execute periodically. It can however, be used to trigger periodic execution instead of using the configuration option. This can be useful if execution is dependent on wall clock time or if other periods not supported by the PERIOD parameter are desirable.

Example: If there are two Control Modules, CM1 and CM2, to be executed every 8 hours, both must be configured with a period of NOPERIOD. Two other Control Modules, CM3 and CM4 are configured to run once a minute (PERIOD = 60s, PHASE = any valid phase) to determine if it is time for CM1 and CM2 to execute. If it is time for them to run, then CM3 and CM4 store to CM1 and CM2's Process Special trigger parameter causing both to execute. CM3 and CM4 can be configured such that they cause CM1 and CM2 to execute at the appropriate time based on either of the following:

- Wall-clock time; for instance, every 8 hours at 8 a.m., 4 p.m., 12 p.m.
- Another CM's activation time; for instance, every 8 hours since CM3 has been activated.

If the two CMs need to execute in a specific order, CM1 followed by CM2, the following options exist:

- CM 3 could be configured to execute in Phase 0 and CM4 in Phase 1, thus causing the triggers to occur 500 ms apart and the subsequent execution order for CM1 and CM2 are maintained.
- CM3 and CM4 can be configured to execute in the same phase with CM3 having a lower ORDERINCEE value than CM4. CM3 would trigger CM1 to execute and then CM4 would trigger CM2. ACE maintains the execution order of the triggered blocks based on the order the parameter store occurred.

- CM3 can request CM1's execution to occur immediately and CM4 could request CM2 to execute in 1 second.

4.5.5 Process special and scaling

The CM parameter SCALEPERIOD is supported to allow a program or another function block to store a scaling value just before the module is to execute. The parameter can only be written if the CM PERIOD is set to NONE. A write to this parameter is rejected if the CM has any other period. This allows blocks with period-related parameters to be scaled appropriately based on the time since the CM's last execution. The CM supplies this value to each of its contained function blocks. The floating-point value written is the number of seconds in the period. The value will be validated against the valid range of between the minimum ACE period and 24 hours.

If this parameter is read while the period is NONE, it will return the last value written to it or the default value of one minute if it has not been written. If the parameter is read while the CM period is something other than NONE, the parameter will return the current period value.

4.6 CEE/ACE Memory and Processing Resources

Related topics

“CEE/ACE processing resources” on page 28
“CEE/ACE memory resources” on page 28
“Typical requirements” on page 29

4.6.1 CEE/ACE processing resources

CEE/CPM and ACE Processing Resources		
Definitions: *PU = Processing Unit. The PU represents a platform independent amount of processing resources (time) required to complete a predefined amount of computational (control) work.* *Maximum Cycle Loading: Over a cycle (0-39), the "Average CPU Used" (CPUCYCLEAVG) statistic is not to exceed the stated maximums.*		
Minimum Reserved CPU to be Maintained During Runtime	**20% - CPM 20% - ACE** (See Note below)	
	PU Maximum	**Maximum Cycle Loading**
500 ms CEE/ACE	15000 PU/sec	60%
50 ms CEE - Non-Redundant Configuration	3600 PU/sec	60%
50 ms CEE - Redundant Configuration	1600 PU/sec	60%
5 ms CEE - (Non-Redundant only)	2400 PU/sec	40%

Attention
CPUFREEAVG is not supported by ACE. CPU Usage from Windows Task Manager provides the %CPU used; ACE reserved is 100% - %CPU being used

4.6.2 CEE/ACE memory resources

CEE/CPM Memory Resources and Block Configuration		
Definition: *MU = Memory Unit = 1 Kbytes = 1024 bytes*		
	CPM	**ACE**
Maximum Available CEE Memory Resources	4000 MU	128,000 MU
Maximum Total Number of CMs, SCMs and IOMs Configurable per CEE	1000	4095* (*IOMs not supported)
Maximum Number of Control Modules with PERIOD of NOPERIOD, Process Special	N/A	N/A
Maximum Number of Component Blocks per CM	100	100
Maximum Total Number of Steps and Transitions (in all handlers) per SCM	400 (200 Step/Transition pairs)	400 (200 Step/Transition pairs)

4.6.3 Typical requirements

The following table identifies typical processing and memory resource requirements for SCMs based on Processing Resources per module and Memory Resource usage. The Processing Resources are computed as "Processing Resource Consumption (PU/module execution) divided by Execution Period (sec/module execution)". Refer to CEE/CPM and CEEACE Memory Resources sections for more information about the terms PU and MU.

Typical Module Types (FB Content in Parenthesis)	Processing Resource Consumption (Per Module)	Memory Resource Usage
Sequence Control Module A (1 each of Main, Hold, Stop and Abort Handlers, 10 Steps with 8 Outputs each, 10 Transitions with 5 Conditions each, 10 Recipe items, 5 History items) SCM has total of 10 Steps and 10 Transitions among the 4 Handlers	2.0	28.9
Sequence Control Module B (1 Main Handler, no other Handlers, 20 Steps with 4 Outputs each, 20 Transitions with 3 Conditions each, 10 Recipe items, 5 History items) SCM has total of 20 Steps and 20 Transitions	2.0	35.7
Sequence Control Module with an alias table size of 45 rows by 100 columns (1 each of Main, Hold, Stop and Abort Handlers, 10 Steps with 8 Outputs each, 10 Transitions with 5 Conditions each interspersed in all the handlers, 10 Recipe items, 5 History items) SCM has total of 10 Steps and 10 Transitions among the 4 Handlers	2.0	128.5
Sequence Control Module with an alias table size of 500 rows by 9 columns (1 each of Main, Hold, Stop and Abort Handlers, 10 Steps with 8 Outputs each, 10 Transitions with 5 Conditions each interspersed in all the handlers, 10 Recipe items, 5 History items) SCM has total of 10 Steps and 10 Transitions among the 4 Handlers	2.0	124.5

4.7 Configuring ACE for CAB

The use of Custom Algorithm Blocks (CAB) and Custom Data Blocks (CDB) affect the ACE node in the following areas:

- "ACE processing requirements for CAB/CDB" on page 30
- "ACE memory utilization for CAB/CDB" on page 30
- "Operations" on page 69 (ACE shutdown to clear CAB memory)

The sections that follow provide the basic considerations to be used in ACE planning and operations. For a complete discussion of the effect of CAB/CDB on ACE, see *CAB and CDB System Planning and Design*.

Related topics

"ACE processing requirements for CAB/CDB" on page 30

"ACE memory utilization for CAB/CDB" on page 30

4.7.1 ACE processing requirements for CAB/CDB

The CPU utilization for CAB and native blocks is covered in *Control Builder Components Theory*. There is no processing requirement for CDB.

4.7.2 ACE memory utilization for CAB/CDB

Refer to the document Determine ACE memory utilization

4.8 Peer-to-Peer Considerations

Related topics

“Peer-to-peer operations using ACE” on page 31
“Peer-to-peer update rates” on page 31
“CEE/ACE communications performance” on page 31
“Timed function block execution” on page 32
“Peer environments and subscription periods” on page 32
“Function block support (CPM/ACE comparison)” on page 33

4.8.1 Peer-to-peer operations using ACE

You can implement peer-to-peer communications among CEEs in multiple Application Control Environment (ACE) supervisory controllers that belong to the same system Server over the Fault Tolerant Ethernet (FTE) or Ethernet network. The ACE supervisory controller can peer-to-peer with CEEs in multiple C200 CPMs through an optional direct connection to the ControlNet supervisory network.

With R410, you can initiate peer-to-peer communication between the ACE-CEE and non-CEE points such as PMD, SCADA, TPS, and Safety Manager points, through FTE network. The peer-to-peer communication between CEE and the Experion server points are licensed using "Experion server Peer Responder" feature.

The basics of Experion peer-to-peer operations can be found in *Basic Peer-to-Peer Design Concepts*.

4.8.2 Peer-to-peer update rates

The configurable peer-to-peer update rates for the ACE are 500 ms; 1 s; 2 s; 5 s; 10 s; and 20 s. The default is 2 s.

4.8.3 CEE/ACE communications performance

The maximum total parameter access response rate for the 500-ms CEE/ACE (including display, Fast/Slow History, Excel I/ODBC Exchange, and peer communications) is 2000 parameters/second (PPS). Peer-to-peer communications performance is summarized below:

CEE to CEE - Peer-to-Peer Communications Performance per CEE	500 ms CEE/ACE
Maximum number of peer-to-peer connections to other ControlNet resident CEE type environments (CPM, FIM, IOLIM)	30
Maximum number of peer-to-peer connections as target initiated by other ControlNet resident CEE type environments (CPM, FIM, IOLIM)	30
Maximum Initiator Node Pull/Get Request Rate - To all target nodes.	500 PPS
(Based on the number of requests for peer data and the peer update rate.)	**ControlNet**
(There are no guarantees that an Ethernet network can assure the timely delivery of critical data packets. Users are cautioned to consider the non-deterministic nature of an Ethernet network before designing and configuring a mission critical control strategy to use it. An Ethernet network is subject to collisions, deferred transmissions, and other anomalies that may delay; the arrival of critical data packets. The results of internal tests show that under a full load condition approximately 1 in 30,000 peer fetches arrive at a peer node at greater than twice the configured peer update rate.)	250 @ 500 ms 500 @ 1 sec 1000 @ 2 sec 2500 @ 5 sec 5000 @ 10 sec
	Ethernet

CEE to CEE - Peer-to-Peer Communications Performance per CEE	500 ms CEE/ACE
	250 @ 500 ms 500 @ 1 sec 1000 @ 2 sec 2500 @ 5 sec 5000 @ 10 sec
Maximum Target Node Response Rate to Pull/Get Requests - From all initiator nodes.	500 PPS
Maximum Initiator Node Push/Store Request Rate - To all target nodes. (The SCM Step and Push are the only block types that can currently initiate peer push/store requests for CEE-to-CEE peer communications.)	50 PPS
Maximum Target Node Response Rate to Push/Store Requests - From all initiator nodes.	50 PPS
Maximum Initiator OPC Pull/Get Request Rate - To all target OPC servers. *(There is no limit imposed on number of different OPC servers that can be accessed by the ACE/CEE.)*	2000 PPS
Maximum Initiator OPC Push/Store Request Rate - To all target OPC servers. (There is no limit imposed on number of different OPC servers that can be accessed by the ACE/CEE.)	500 PPS

4.8.4 Timed function block execution

Since the ACE/CEE runs on a time-sharing operating system, the timed event for function block execution may be delayed or interrupted by other tasks. The ACE/CEE function block includes statistical parameters for calculating timed event for function block execution.

The time is measured from the previous cycle's start of FB execution to the current cycle's start of FB execution. Minimum, Average, and Maximum statistics are provided. Ideally, all three values would be equal to the Base Execution Period (500 msec for ACE).

The parameters are:

BASPERIODMIN	Minimum Time Between Execution Cycles
BASPERIODAVG	Average Time Between Execution Cycles
BASPERIODMAX	Maximum Time Between Execution Cycles

4.8.5 Peer environments and subscription periods

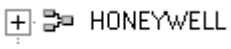
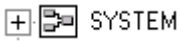
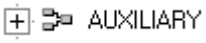
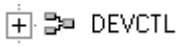
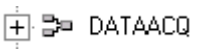
You can identify different peer environments and configure individual subscription periods and store response time values through the applicable CEE configuration form in Control Builder. This includes other ACE supervisory controllers, C200 Process Controllers, and external servers, such as OPC servers. The configuration form also specifies default subscription period and store response time values to be used for all peer environments without individually specified settings. The individual parameters are described below. In addition to system wide default values; the values for specific CEE peers can be adjusted by users with an Engineer access level or higher in the Monitor mode of Control Builder.

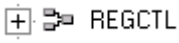
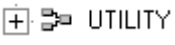
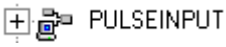
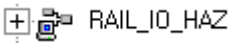
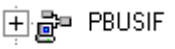
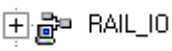
Parameter	Description
All Environments	

Parameter	Description
SUBSCPERIOD	The Peer Update Rate (SUBSCPERIOD) parameter is the rate at which this CEE subscribes to data from other CEEs through peer-to-peer communications. In the event of a peer-to-peer communications problem such as a broken cable, the connection will timeout, returning an error
STRRESP	Store Response - Specifies the default store response expiration time to be used for all peer environments in seconds.
Individual Environments	
PEERENV[]	Peer Environment - Specifies the peer environment that is to have specific peer subscription period (PEERSUBSCPER[ ]) and peer store response time (PEERSTRRESP[ ]) values configured for it.
PEERSUBSCPER[ ]	Peer Subscription Period - Specifies the specific update period to be used for the corresponding peer environment number (PEERENV[ ]) in seconds.
PEERSTRRESP[ ]	Peer Store Response Time - Specifies the specific store response expiration time to be used for the corresponding peer environment number (PEERENV[ ]) in seconds. If a store response is not received within the configured expiration time, the store is marked failed

4.8.6 Function block support (CPM/ACE comparison)

Since ACE controller operation is based upon the Control Processor Module (CPM) design, ACE supports many of the same function blocks as the CPM. The ACE controller does not support any of the existing function blocks associated with I/O communication interface. The CPM does not support the new UCNIF block. The following table summarizes the Function Block Libraries in Control Builder and identifies which control environment supports them.

If Function Block Is From This Library in Control Builder . . .	Then, It Can Be Used With This Control Environment . . .
 (Typical Fieldbus Device vendor library.)	CPM/CEE
	ACE/CEE CPM/CEE (Restrictions apply - loaded strategies must contain supported blocks only.)
	ACE/CEE CPM/CEE
	ACE/CEE CPM/CEE
	ACE/CEE CPM/CEE
	CPM/CEE
	CPM/CEE
	CPM/CEE

If Function Block Is From This Library in Control Builder . . .	Then, It Can Be Used With This Control Environment . . .
 LOGIC	ACE/CEE CPM/CEE
 REGCTL	ACE/CEE CPM/CEE
 SCM	ACE/CEE CPM/CEE
 UTILITY	ACE/CEE CPM/CEE
 PULSEINPUT	CPM/CEE
 RAIL_IO_HAZ	CPM/CEE
 PBUSIF	CPM/CEE
 RAIL_IO	CPM/CEE
 FBUSIF	CPM/CEE
 EXCHANGE	CPM/CEE
 UCNIF	ACE/CEE

5 LCN Prefetch for ACE CMs

You *cannot* use Local Control Network (LCN) prefetch with Sequential Control Modules (SCM) and Recipe Control Modules (RCM) in ACE. These modules use the dynamic fetch functionality to optimize peer data transfers.

Related topics

“About LCN prefetch” on page 36
“Functional overview of prefetch request” on page 37
“CEE Hold Breath interaction with prefetch” on page 38
“Block Process Special (BPS) interaction with prefetch” on page 40
“Configuring CMs for Prefetch” on page 41
“ACE CM with Prefetch execution scenario” on page 44
“Prefetch Checkpoint considerations” on page 45
“Prefetch Migration and Import/Export considerations” on page 46
“Viewing Prefetch statistics in OPC Gateway” on page 47

5.1 About LCN prefetch

The prefetch function reduces the number of peer parameter data transfers between the ACE and the LCN and still guarantees the freshness of peer parameter data. Beginning in Experion R311, the ACE CMs using prefetch reduce the frequency of peer data requests by allowing function blocks to request (or prefetch) their input data shortly before the data is actually needed during block execution rather than at a continuously subscribed periodic (publish/subscribe) rate.

Tip

Prefetch is useful for efficient data transfers in slow executing CMs. For typical periods of 1 or 2 seconds, the frequency of peer data requests will not be greatly reduced.

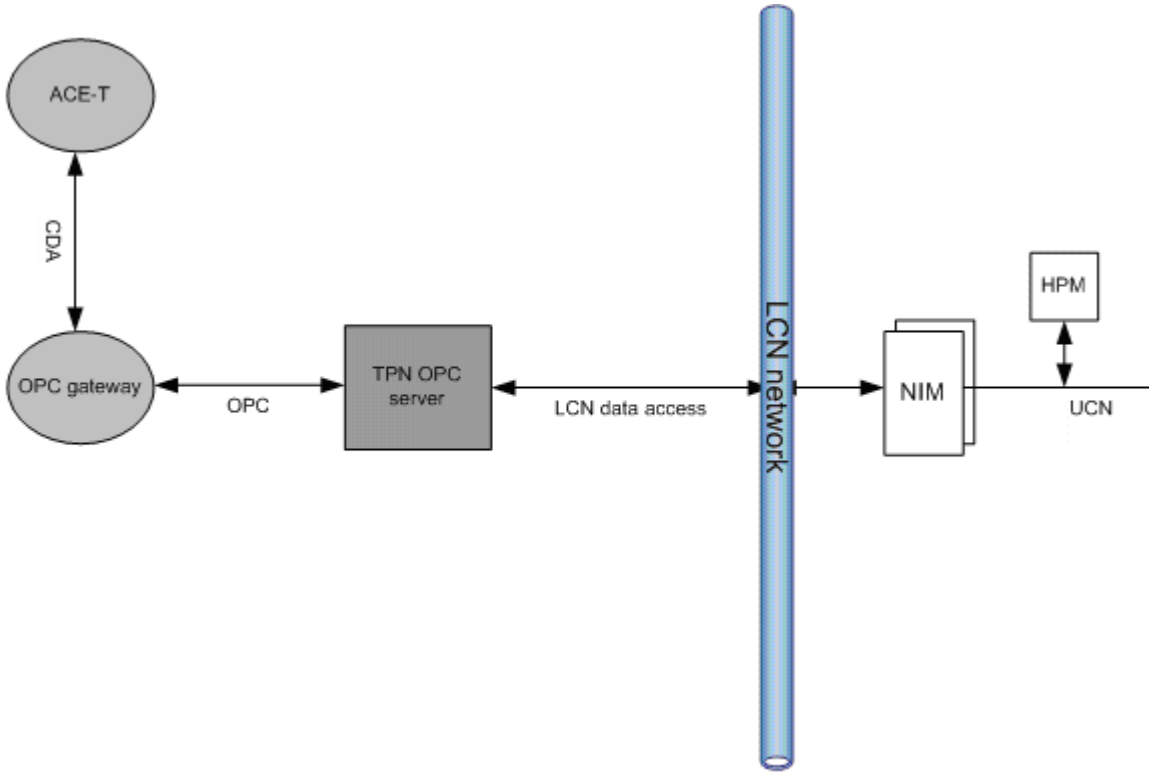
5.2 Functional overview of prefetch request

As shown in the following illustration, a prefetch request from the ACE-T node is sent through the Control Data Access (CDA) to the OPC gateway. The OPC gateway forwards the request to Total Plant Network (TPN) OPC server using the OPC interface. The TPN OPC server forwards the request to the intended node, which could be owned by the HPM module. The node responds to the requests to the TPN server and the server forwards the request to the OPC gateway. The OPC gateway sends the response through the CDA to the blocks running in the ACE-T node.

Attention

In Experion R311, you can only use the prefetch functionality with OPC gateways connected to the LCN that have a ProgID of hci_tpnserver.

Table 9: Typical path for LCN prefetch request



5.3 CEE Hold Breath interaction with prefetch

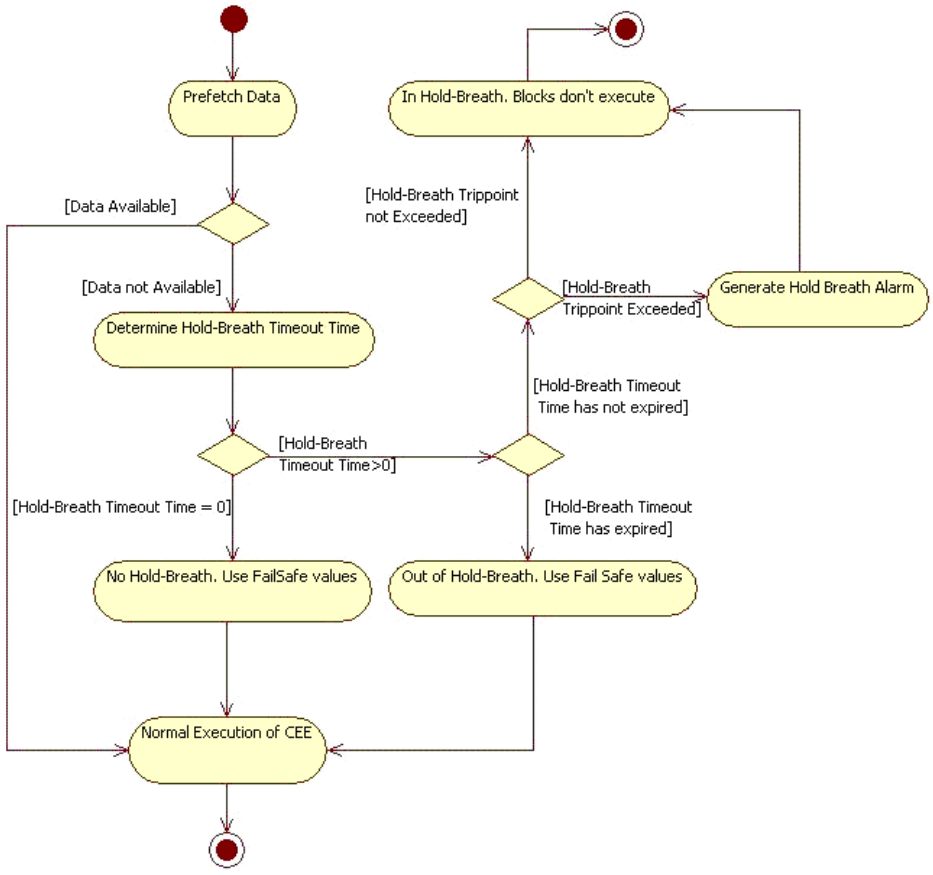
The term "Hold Breath" refers to stopping normal execution of scheduled function blocks in an ACE CEE because peer prefetch data is not available for execution of an ACE Control Module configured for prefetch. An ACE Control Module configured for prefetch triggers collection of peer data in anticipation that peer data will be available during execution of the ACE Control Module.

The CEE stops execution (hold's breath) of **all** (with and without prefetch) control strategies, when prefetch data is not available prior to scheduled execution for an ACE Control Module. Communication delays long enough to cause the CEE to Hold Breath is rare and usually associated only with either a swap or switchover of a server node.

When a hold breath does occur, the CEE holds execution of all its blocks regardless of whether its PEERREADOPT parameter setting is PREFETCH or PUBSUB. This means all Control Modules along with all SCM, RCM, and UCM in the control strategy will stop executing for the duration of the Hold Breath timeout time. The CEE does not attempt to recover the scheduling time lost due to the occurrence of Hold Breath.

The maximum timeout value for Hold Breath is configurable using the HOLDBRTHTM parameter on the Main tab of the configuration form for the ACE block. This is the hold-breath timeout time after which CEE continues execution using fail safe data for the ACE Control Modules without their peer data, as illustrated in the following diagram. This is similar to the TPS-AM Hold Breath mechanism. See the “Configuring CMs for Prefetch” on page 41 section for more information on configuring the HOLDBRTHTM parameter.

Table 10: Typical CEE Hold Breath scenario



Attention

The CEE differentiates between a communication failure and a failure in obtaining a response from the peer.

If the CDA responds to a prefetch request with an error indicating a communication failure, the blocks continue executing with the error status.

CEE Hold Breath happens when **no response**, neither data nor error status, is received from the peer within the configured Prefetch Anticipation Time (PAT).

CEE Hold Breath is applicable only to ACE CEE's which contain control modules configured with PEERREADOPT of PREFETCH. Under no circumstances, can Hold Breath occur in C200/C300 CEE.

5.3.1 Occurrence of Hold Breath when an HPM point is deleted

When the ACE is subscribed to an HPM point that is deleted, the ACE may report Hold Breath events as long as it attempts to access the deleted point. The Hold Breath occurrence increases significantly until the HPM point is re-loaded.

In such a scenario, you must remove all ACE references to the deleted point.

5.3.2 Occurrence of Hold Breath during NIM/HG/AM node failover

It is possible that the pre-fetch requests from ACE targeted to LCN data owners may get lost when LCN data owners like NIM/HG/AM failover. During this time, the responses for the pre-fetch requests are not received at the ACE. As a result, Hold Breath can occur on the ACE node.

In such a scenario, Hold Breath events occur until the data arrives or until the Hold Breath timeout occurs; whichever is earlier.

5.3.3 CEE Hold Breath alarm indication

As shown in the following example illustration, when a Hold Breath occurs, a CEE Hold Breath alarm is generated in the System Status display in Station.

Hold Breath Alarm and Read Freeze after removal of LCN Cable

When one of the LCN (A or B) is removed, the response time for the pre-fetch requests may be delayed. Also, the response time may be more than 3 seconds. In such a scenario, Hold Breath may occur if the PAT (Prefetch Anticipation time) on ACE is configured as less than 3 seconds.

In such a scenario, Hold Breath events occur until the data arrives or until the Hold Breath timeout occurs; whichever is earlier.

5.3.4 Prefetch Anticipation Time (PAT)

The Prefetch Anticipation Time or PAT is the time in seconds prior to the execution of a CM that the request for peer data collection is issued. This value is configurable through the PFANTICTIME parameter on the Main tab of the configuration form for the ACE block. See the “Configuring CMs for Prefetch” on page 41 section for more information on configuring the PFANTICTIME parameter.

5.4 Block Process Special (BPS) interaction with prefetch

ACE Control Modules configured as BPS with PEERREADOPT of PREFETCH must trigger their peer data collection prior to the execution of the ACE Control Module. The CEE provides an opportunity for an ACE BPS Control Module configured with prefetch to trigger data collection of its peer references. The ACE BPS Control Module with prefetch does **not** execute until either the requested peer data becomes available or the CEE times out waiting for the peer data. Then, the ACE BPS Control Module uses fail safe values for its peer prefetch references.

- The actual execution of BPS Control Modules may happen based on the configured PAT (PFANTICTIME) parameter in seconds after the actual trigger for BPS.

5.5 Configuring CMs for Prefetch

The Peer Read Option (PEERREADOPT) setting for the Control Module determines whether configured peer connections will initiate prefetch (PREFETCH) requests to the target or will work with publish/subscribe (PUBSUB) logic.

You can have a mix of Control Modules in which each individual CM can have its PEERREADOPT value configured as either PREFETCH or PUBSUB. For example, consider a scenario where CM1 is interdependent on CM2 and CM3 and CM1's PEERREADOPT value is configured as PREFETCH. In such a case, CM2 and CM3 can have their PEERREADOPT value configured as either PREFETCH or PUBSUB irrespective of CM1's PEERREADOPT value.

The PEERREADOPT is visible only when the Control Module is assigned to a CEEACE block.

The PEERREADOPT is not available on SCMs, UCMs and RCMs, since they do **not** have the prefetch capability.

You can only configure the value for the PEERREADOPT parameter in the **Project** mode before loading the CM. Any change in value requires a reload of the CM.

Control Modules must have a PERIOD setting greater than the Prefetch Anticipation Time (PFANTICTIME) value to allow the PEERREADOPT value of PREFETCH. Otherwise, during the CM load, the PEERREADOPT selection of PREFETCH is clamped to PUBSUB, if the CM PERIOD is less than or equal to the ACE PFANTICTIME value.

For loaded CM with PEERREADOPT of PREFETCH, you are not allowed to change CM PERIOD to a value less than or equal to the ACE PFANTICTIME value.

Load fails for a CM configured with the PEERREADOPT selection of PREFETCH, if it has connection references to Peers that do not support prefetch, such as a C300 Server.

You enable or disable the CEE Hold Breath based on the value set for the Hold Breath Timeout Time (HOLDBRTHTM) parameter in seconds as listed in the following table.

If HOLDBRTHTM Value Is . . .	Then, CEE Hold Breath Is . . .
Zero (0)	Disabled. If no response is received for prefetch requests within the Prefetch Anticipation Time, then the blocks execute immediately with fail safe values. CEE does not hold execution of blocks.
Greater than Zero (0)	Enabled. If no response is received for prefetch requests within the Prefetch Anticipation Time, then the CEE holds execution of all blocks until the data/error is received or Hold Breath time out happens. The CEE holds execution of blocks after the completion of Prefetch Anticipation Time for the number of seconds configured by the HOLDBRTHTM parameter. After Hold Breath timeout occurs, the blocks are executed with fail safe values.

You can configure the HOLDBRTHTM value in the granularity of 0.5 seconds up to seven seconds. For example, the possible values can be 0, 0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5, 5.5, 6, 6.5, and 7.

You can configure the PFANTICTIME value in the granularity of 0.5 seconds up to 10 seconds. For example, the possible values can be 1, 1.5, 2, 2.5, 3, 3.5, 4...10.

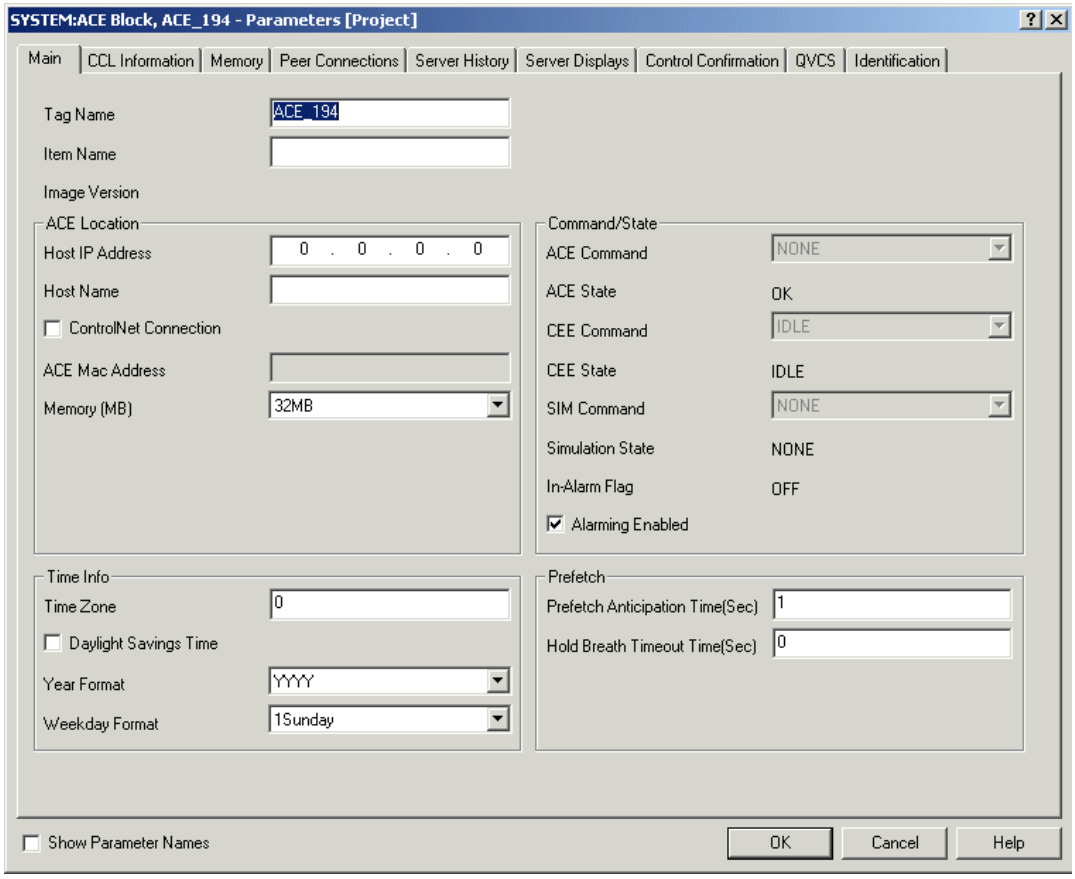
The following procedure assumes that ACE and Control Module blocks have been created and only deals with configuring the LCN Prefetch related parameters. All illustrations are for example purposes only and do not represent valid configurations.

Prerequisites

- Control Builder is running and you are logged on with sufficient privileges to create a control strategy.
- You have created an ACE/CEEACE block.
- You have created Control Modules to be assigned to ACE.
- You have created an OPC Gateway block with an LCN connection.

To configure ACE CM for prefetch

1 In the **Project** tab, double-click the ACE block in the root directory to open its Module Properties (Configuration) form.



2 On the **Main** tab, do you want to change the default Prefetch Anticipation Time value of 1 second?

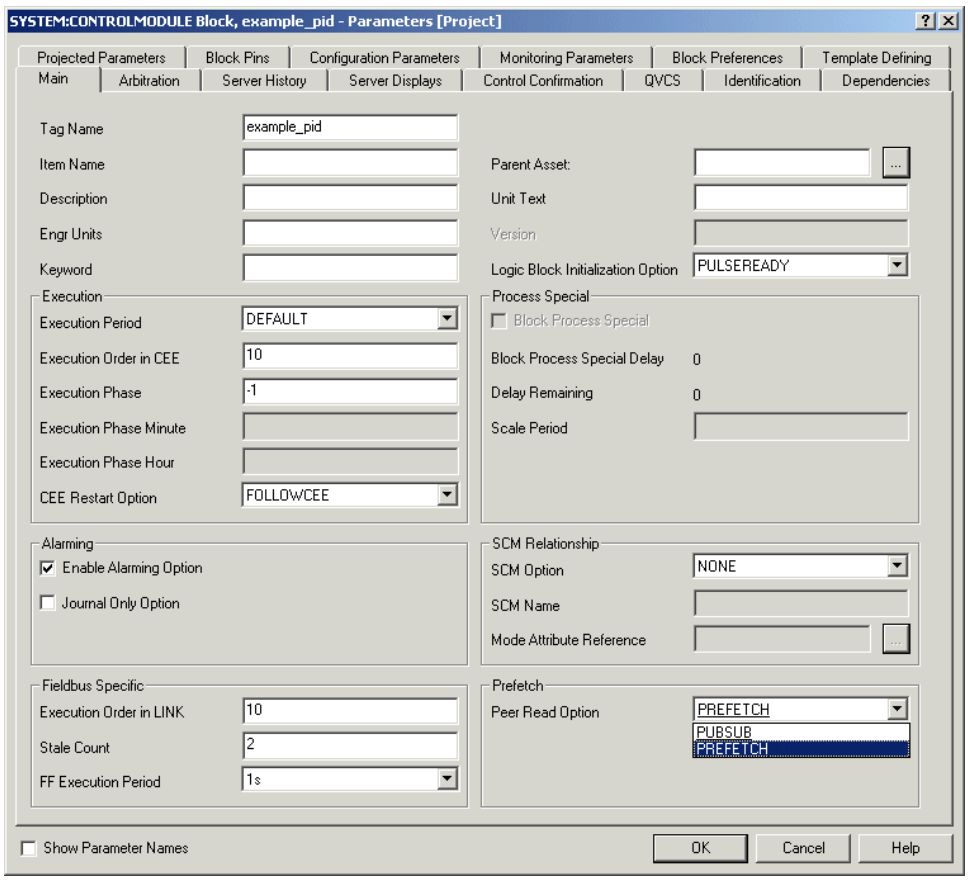
- **Yes**: Click the cursor in the **Prefetch Anticipation Time(Sec)** box and key in the desired value up to 10 seconds. Press the **Enter** key.
- **No**: Go to the next Step.

3 Do you want to change the default Hold Breath Timeout Time of 0 (zero), which means CEE Hold Breath is disabled?

- **Yes**: Click the cursor in the **Hold Breath Timeout Time(Sec)** box and key in the desired value up to 7 seconds, which also enables CEE Hold Breath. Press the **Enter** key.
- **No**: Go to the next Step.

4 Click the **OK** button to close the form and save the settings.

5 In the **Project** tab, right-click the desired Control Module either in the directory or CEEACE and select **Module Properties** from the list to open its configuration form.



6 On the **Main** tab, click the down-arrow button in the right side of the **Peer Read Option** box and select **PREFETCH** from the list.

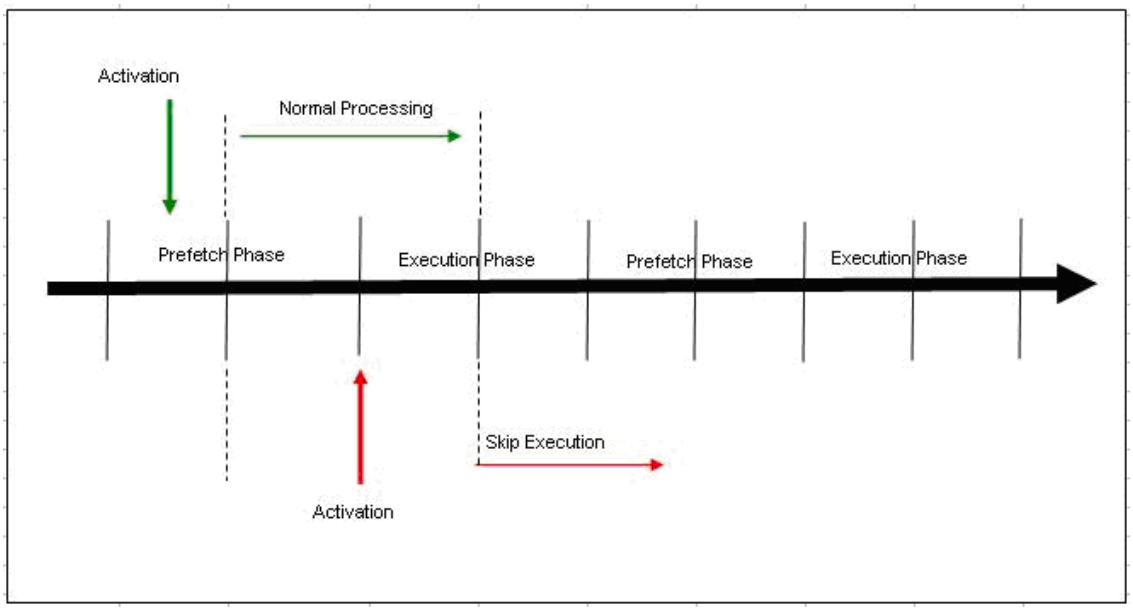
7 Click the **OK** button to close the form and save the changes.

8 This completes the prefetch configuration for ACE CM. Repeat Steps 5 to 7 for other CMs, as required.

5.6 ACE CM with Prefetch execution scenario

The prefetch processing is only triggered when CEE state is RUN and CM state is ACTIVE. A CEE IDLE to RUN transition or a CM INACTIVE to ACTIVE transition may happen at any point in time. There are two possible scenarios, as illustrated in the following diagram and described below.

Table 11: Execution of a CM with Prefetch



- CM gets activated before the Prefetch Phase (Execution Phase - PAT in phases)
  - No special handling is required, normal prefetch and block execution will happen.
- CM gets activated after the Prefetch Phase (Execution Phase - PAT in phases) but before the Execution Phase.
  - In case of prefetch, the CM needs to get a chance to trigger peer data collection prior to execution.
  - The CEE gives the CM a chance to trigger peer data collection prior to execution by skipping the first execution of the CM.
  - In the next prefetch phase, the CM gets a chance to trigger peer data collection. Normal CM and prefetch processing continues.
  - The first execution of the CM may happen PERIOD + PFANTICTIME after CEE transition to RUN or CM transition to ACTIVE.

5.7 Prefetch Checkpoint considerations

The configured values of the parameters PFANTICTIME and HOLDBRTHTM will be restored on a Checkpoint Save-Restore. On a checkpoint restore, the CEE state will be IDLE. The Prefetch requests will not be processed, when CEE is IDLE.

5.8 Prefetch Migration and Import/Export considerations

ERDB migration will happen as follows:-

- On ERDB migration of a pre R311 database, ACE HOLDBRTHTM will be set to 7 seconds.
- On ERDB migration of a pre R311 database, ACE PFANTICTIME will default to 1 second.
- On ERDB migration of a new ACE created in R311, the parameters PFANTICTIME and HOLDBRTHTM will retain their values.
- For a new ACE created in R311, PFANTICTIME will default to 1 second and HOLDBRTHTM will default to 0 seconds.

Import of an ACE created in a pre R311 release will have:-

- PFANTICTIME default of 1 second
- HOLDBRTHTM default of 7 seconds.

Import of an ACE created in R311 or later will retain the PFANTICTIME and HOLDBRTHTM parameter values.

5.9 Viewing Prefetch statistics in OPC Gateway

Prerequisites

- Control Builder is running.
- You have loaded ACE with prefetch Control Modules.
- You have loaded OPC Gateway.

To view Prefetch statistics

1. In the **Monitoring** tab, double-click the OPC Gateway block to open its configuration form.
2. Click the **Statistics** tab, to bring it to the front.
3. Check the parameters listed in the **Pre-fetch Statistics** category for real time values.
4. Click the **OK** button to close the form.
5. This completes the procedure

6 Installation and Upgrade

Refer to the *Software Installation and Upgrade Guide* for further details.

7 Configuration

Related topics

“Creating an ACE and CEE” on page 52
“Application Control Environment Block” on page 60
“Control Execution Environment Application Control Environment Block” on page 63

7.1 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. The following error message (2299) is displayed if you try to directly connect the ACE/SIMACE to I/O. "initiator cannot establish connection with the responder" Therefore, ensure that your connection is defined as a 1756 channel.

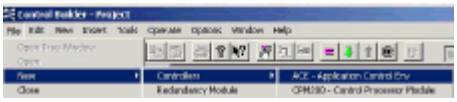
The CEE supports execution of a set of Function Blocks for solving control applications.

It runs in the ACE supervisory controller as a software layer built on top of the control software infrastructure.

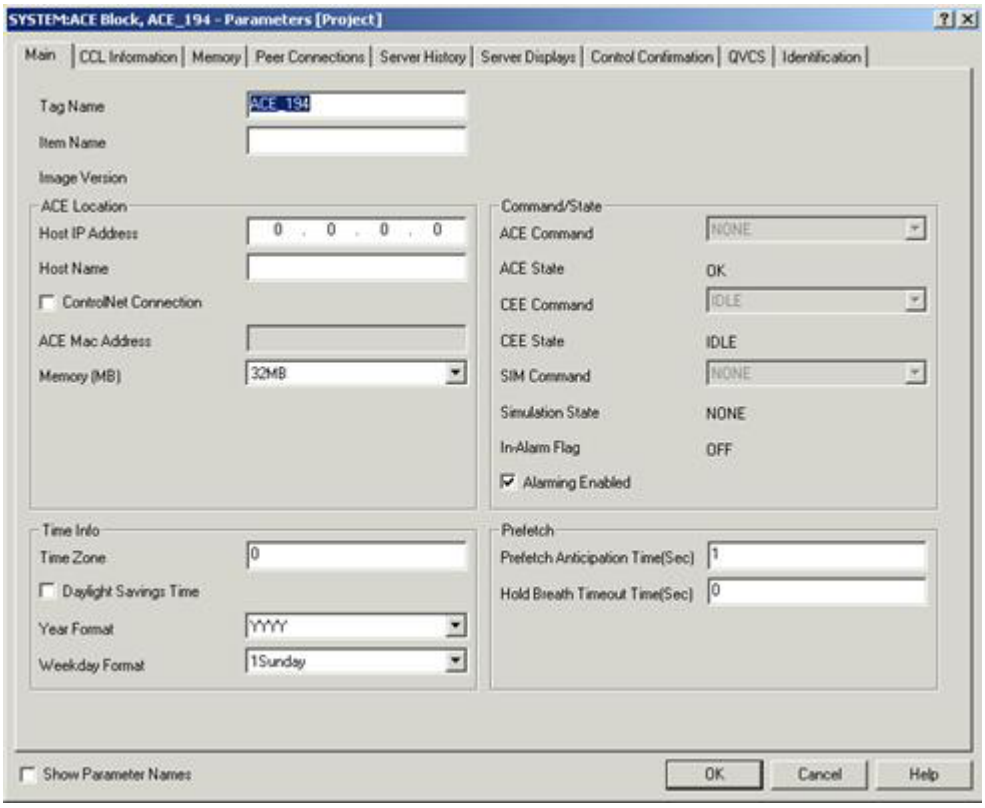
Details of the contents and usage of the Application Control Environment (ACE) Block can be found at “Application Control Environment Block” on page 60. The details and usage of the Control Execution Environment (CEE) block can be found at “Control Execution Environment Application Control Environment Block” on page 63

The following procedure assumes that Control Builder is running and two tree windows are open. All illustrations used in the procedure are for example purposes only.

1 Click **File** > **New** > **Controllers** > **ACE -** .
“Application Control Environment Block” on page 60



Calls up the ACE Block configuration form with Name field highlighted.



2 Key in desired name of up to 16 characters or accept the default. Press <Tab>
Moves cursor to 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. 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 10.

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 Memory (MB) field.

9 Click the down-arrow button to select the desired memory size in megabytes (MB) or accept the default size of 32MB. Press <Tab>.

Moves cursor to the Alarm Enabled selector.

10 The default is alarming enabled. Remove the check to disable alarming. Press <Tab>

Moves cursor to the Time Zone 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 website, if you have a question about the appropriate offset value for your given location.

11 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.

12 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.

13 Accept default or click down-arrow button and select desired format from the list. Press <Tab>.

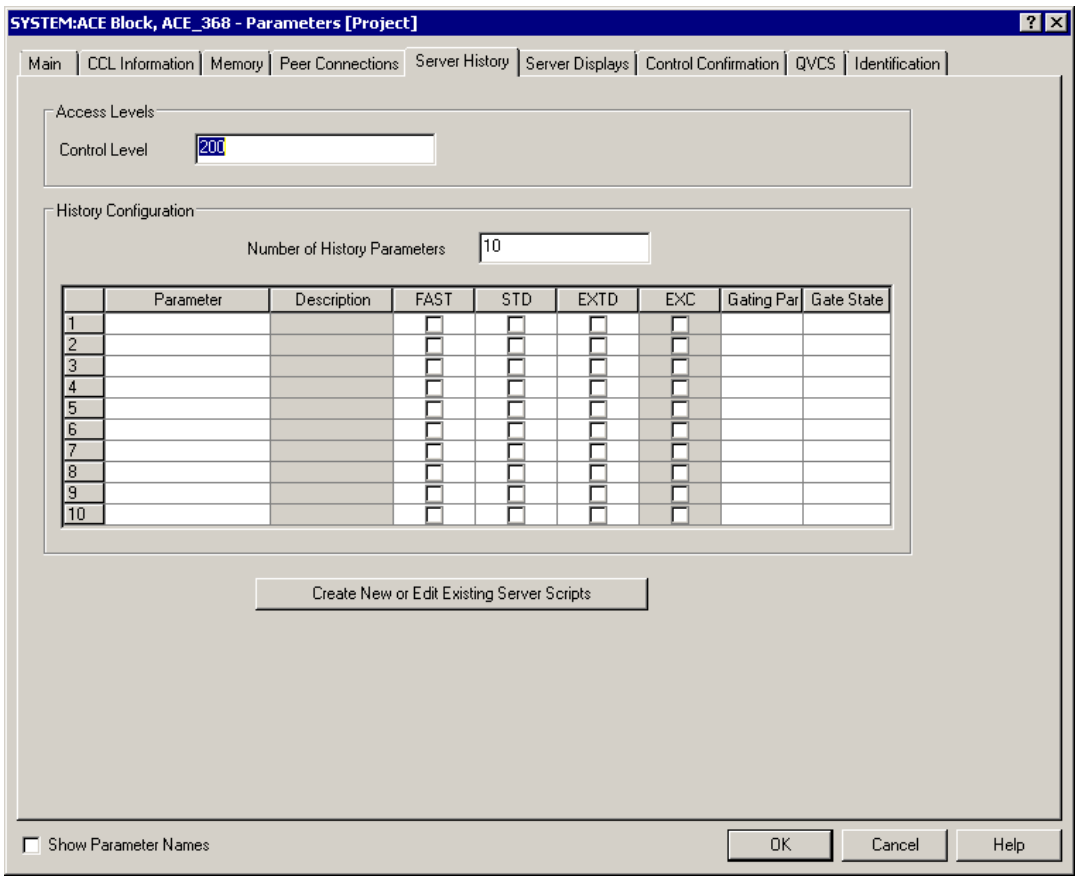
Moves cursor to Weekday Format field.

14 Accept default or click down-arrow button and select desired weekday format from the list. Click cursor in Prefetch Anticipation Time(Sec) field.

Moves cursor to Prefetch Anticipation Time(Sec) field.

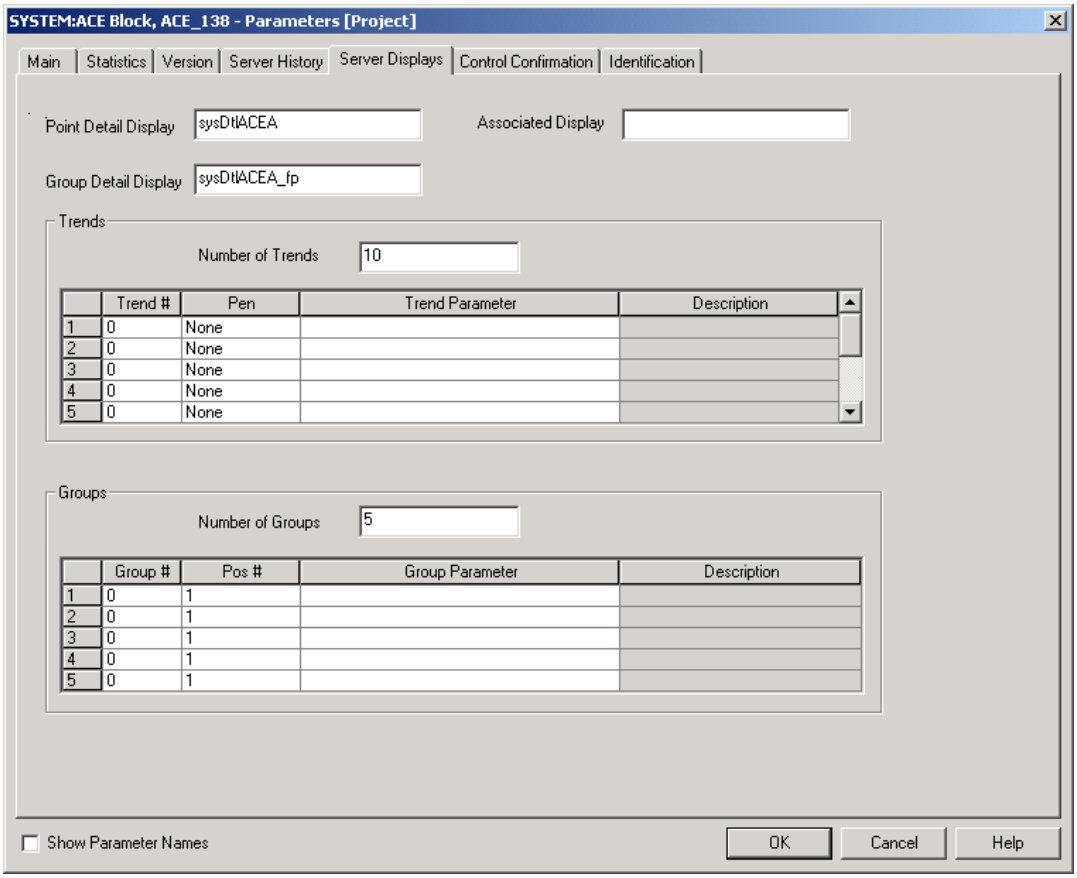
15 See the “Configuring CMs for Prefetch” on page 41 section for more information about configuring Prefetch parameters: Prefetch Anticipation Time (PFANTICTIME) and Hold Breath Timeout Time (HOLDBRTHTM). Click the **Server History** tab.

Calls up the Server History configuration form.

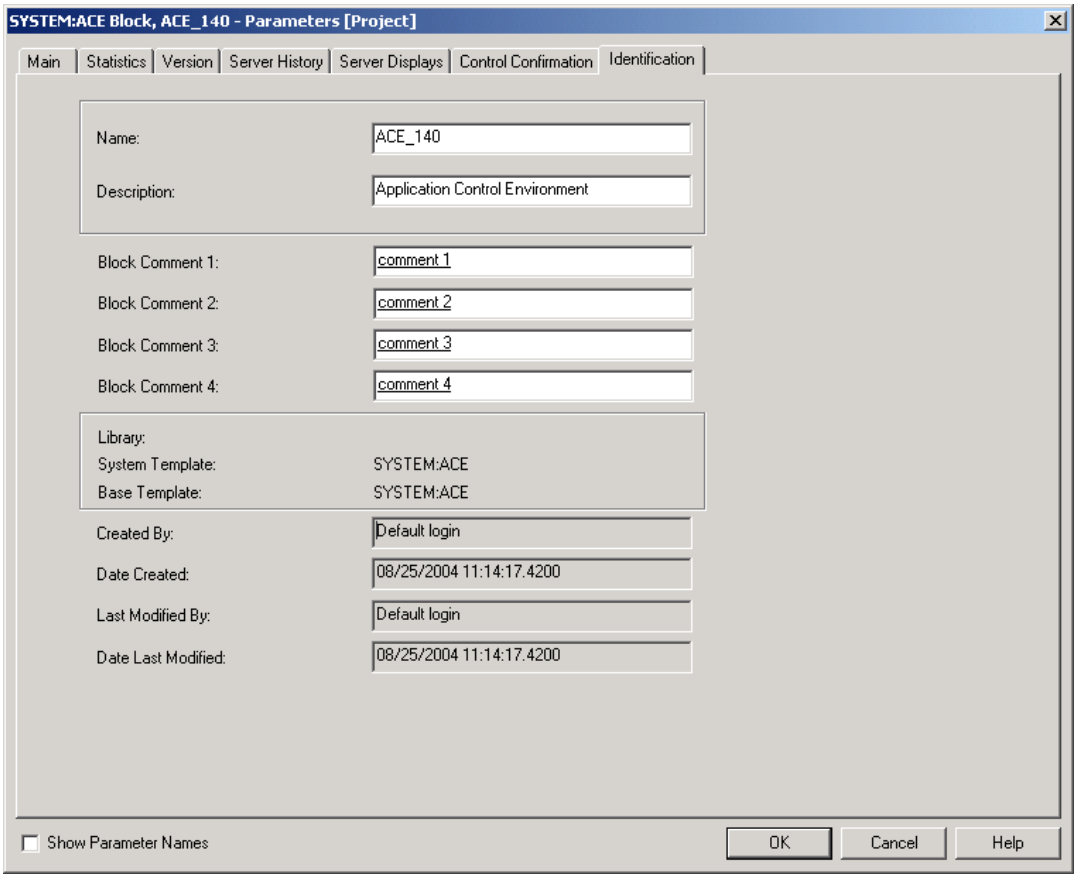


16 Use the on-line help as a guide to complete the configuration entries on this tab. Click the **Server Displays** tab.

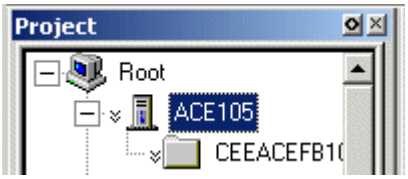
Calls up the Server Displays configuration form.



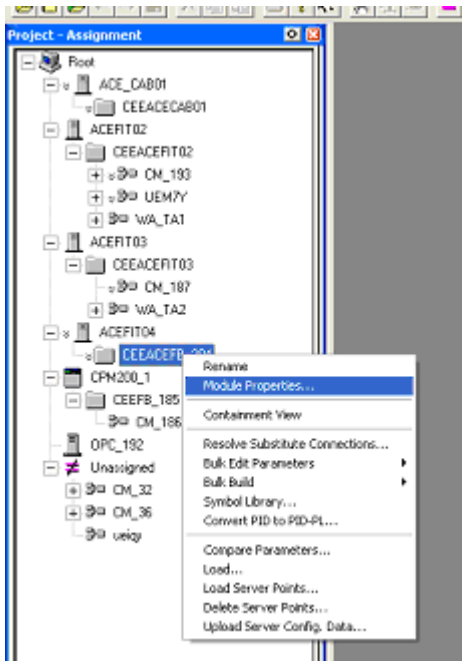
17 Use the on-line help as a guide to complete the configuration entries on this tab. Click **Identification** tab. Calls up the Identification configuration form



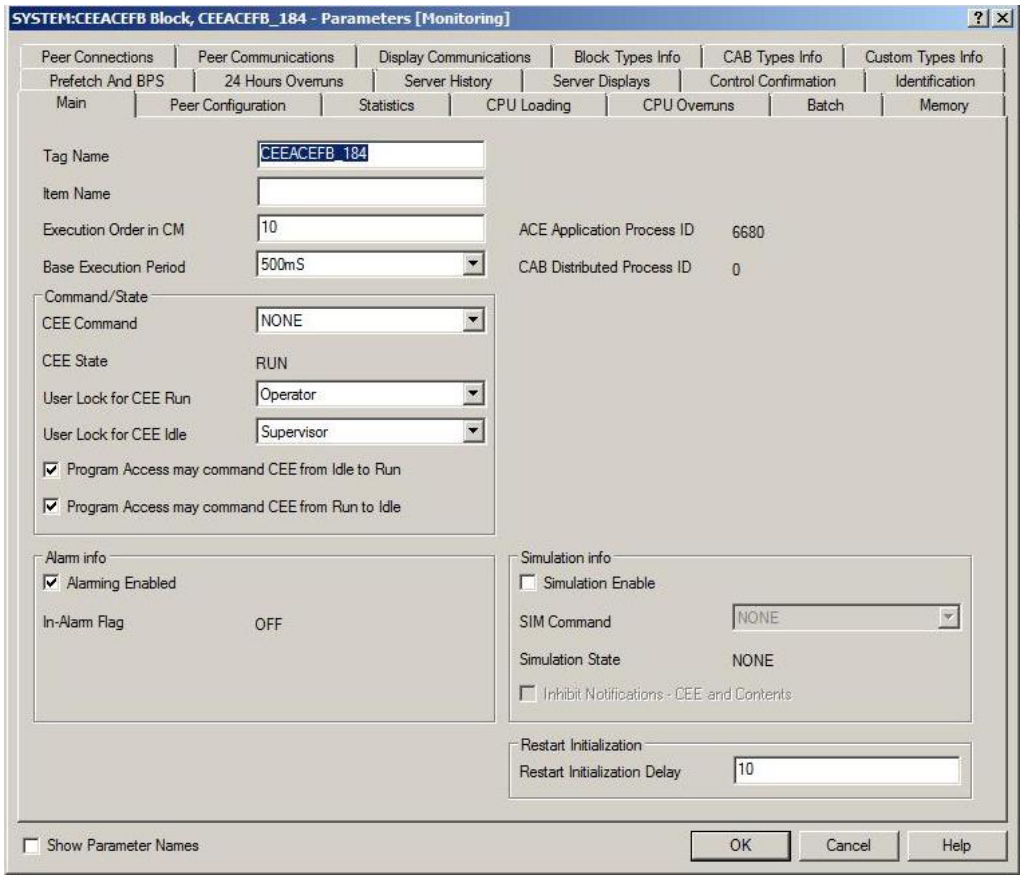
18 Use the on-line help as a guide to complete the configuration entries on this tab. Click the **OK** button.
Closes the form and creates ACE/CEE block icons in Project tab.



19 Right-click CEEACE block icon.
Calls up shortcut menu.



20 Click Module Properties.
Calls up “CEEACE Block” on page 63 configuration form.



21 Key in desired name of up to 16 characters or accept the default. Press <Tab>

Moves cursor to the Item Name field

22 If the CEE is part of an Enterprise Model, enter its Item Name here. Press <Tab>
Moves cursor to Base Execution Period field.

23 Accept the default. Press <Tab>.
Moves cursor to Subscription Period field.

24 Accept the DEFAULT or click down-arrow button and select desired period. Press <Tab>.
Moves cursor to Store Expiration Time field.

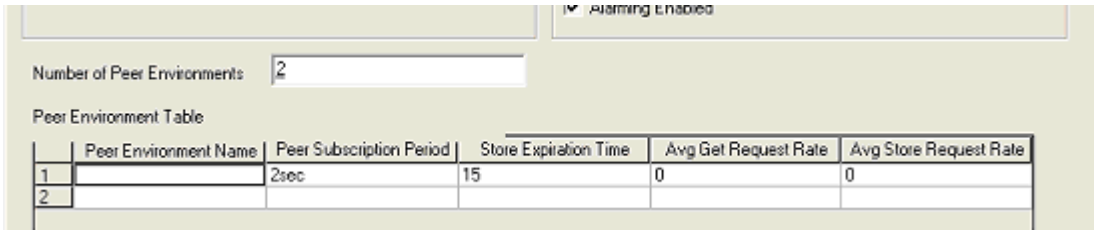
25 Accept the default or key in desired value. Press <Tab>.
Moves the cursor to the Simulation Enable selector.

26 Select if this node is to be used as a SIM ACE (see the *SIM ACE User Guide* for additional details). Press <Tab>.
Moves cursor to Alarming Enabled field.

27 The default is alarming enabled. Remove the check to disable alarming.

28 If peers are to be configured, move to the Number of Peer Environments field and go to item 25. Otherwise, click on the Server History tab and go to Step 32

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.

29 Key in number of peer environments for this ACE. Press <Tab>. Or, skip this field, if no peer environments will be used, click the **Server History** tab and go to Step 31
If peer environments will be used, cursor moves to the Peer Environment Name column in the Peer Environment Table.

30 Key in valid name for existing peer environment. Press <Tab>.
Moves cursor to Peer Subscription Period column.

31 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.

32 Accept the default value of 15 or enter a non-zero value. Return to Step 28 to enter data for another peer, or click the **Server History** tab and go to step 31

33 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.

34 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.

35 This completes the ACE/CEE creation procedure.

7.2 Application Control Environment Block

Related topics

“ACE Block” on page 60

7.2.1 ACE Block

Description	The ACE controller mirrors the basic operations of a Control Processor Module (CPM). It provides the additional capability of communicating with OPC Servers through a Fault Tolerant Ethernet (FTE) or redundant or non-redundant Ethernet network. The ACE program runs on a server grade personal computer using a Windows 2000 Server operating system. Users can optionally connect an ACE controller directly to a supervisory Fault Tolerant Ethernet (FTE) or ControlNet network to support peer-to-peer communications with a C200 process controller. It executes once every 2 seconds.
Function	Serves as an OPC client for communications with OPC servers. Supports peer-to-peer with other ACE controllers connected to the same Server over Fault Tolerant Ethernet or Ethernet network through associated services. The ACE controller that has a direct connection to the supervisory ControlNet can peer-to-peer with C200 controllers that belong to the same Server. Starting with Experion R430, supports peer-to-peer communication between ACE Controller and EHPM Controller.
Inputs	Transmission Control Protocol/Internet Protocol (TCP/IP) Integrated Control Protocol (ICP) communications
Outputs	See above.

Parameters	ACEMACADDR	NAME
	ALMENBSTATE	NTOTMEMDESC
	AUXDESC[0..10]	NUMACEINCON
	BLCKCOMMENT1	NUMACEOUTCON
	BLCKCOMMENT2	NUMCPMINCON
	BLCKCOMMENT3	NUMCPMOUTCON
	BLCKCOMMENT4	NUMEXTBLKS
	CCLCNT	NUMFIMINCON
	CCLINFO	NUMFIMOUTCON
	CCLLOADSTAT	NUMFREEBLKS
	CCLNAME	NUMFREEDESC
	CCLPACKET	NUMIOLMINCN
	CEECOMMAND	NUMIOLMOUTCON
	CEESTATE	NUMREGDESC
	CNETCONNECT	NUMSCEINCON
	CPMCOMMAND[0..NUMCHANS-1]	NUMSCEOUTCON
	CPMSTATE	NUMSIGS
	CPUFREEAVG	NUMSIOLMINCN
	CPUFREEMIN	NUMSIOLMOUCN
	CTRLCONFIRM	NUMUSEDBLKS
	CURTIME	NUMUSEDDESC
	DAY	ORDERINCM
	DAYLIGHTTIME	PFANTICTIME
	DESC	PRIMARYSIG
	ENBMEMALMFL	REASONSET
	FREEMEM	SCANASSOCDSP
	FREEMEMINK	SCANCTRLLVL
	GROUP.NUMPARAMS	SCANGRPDTL
	HIST.NUMPARAMS	SCANPNTDTL
	HOLDBRTHTM	SECOND
	HOSTIPPRI	SECONDARYSIG
	HOSTNAMEPRI	SECSIGSECLVL
	HOUR	SIMCOMMAND
	IMAGEVER	SIMSTATE
	INALM	STATSRESET
	MAXFREEBLKSZ	TIMEZONE
	MAXFREEINK	TOTALMEM
	MINUTE	TOTALMEMINK
	MONTH	TREND.NUMPARAMS
		USEDMEM
		USEDMEMINK
		WEEKDAY

		WEEKDAYFMT YEAR YEARFMT
References	Refer to the *Control Builder Parameter Reference* manual for definitions of each parameter.	

7.3 Control Execution Environment Application Control Environment Block

Related topics

“CEEACE Block” on page 63

7.3.1 CEEACE Block

Description	Provides control functionality for associated Application Control Environment block. This block's parameters characterize the CEE within the ACE controller. This block always runs at an execution period of 2 seconds.
Function	Publishes parameters describing the status and configuration of the CEE. Processes the computation of statistical parameters and notification reporting. Runs on the Windows 2000 operating system. Supports configurable subscription rate for peer-to-peer communications. Supports configurable subscription rate and store response time for specific peer environment. Supports peer-to-peer communications among CEEs assigned to CPMs located in the same management domain. Starting with Experion R430, supports peer-to-peer communication between ACE Controller and EHPM Controller.
Inputs	Transmission Control Protocol/Internet Protocol (TCP/IP) Integrated Control Protocol (ICP) communications
Outputs	See above.

Parameters	ALMENBSTATE	NAME
	BASEPERIOD	NOTIFINHIBIT
	BASEPERIODAVG	NUMACCRQUAVG
	BASEPERIODMAX	NUMACCRQUMAX
	BASEPERIODMIN	NUMACEINCON
	BLCKCOMMENT1	NUMACEOUTCON
	BLCKCOMMENT2	NUMBLKTYPES
	BLCKCOMMENT3	NUMCCLRQU
	BLCKCOMMENT4	NUMCPMINCON
	BLKTYPCOUNT	NUMCPMOUTCON
	BLKTYPDESC	NUMEHPMINCON
	BLKTYPHELPTXT	NUMEXCRQUAVG
	BLKTYPLIB	NUMEXCRQUMAX
	BLKTYPSIZE	NUMEXCRSPAVG
	CDISPAVGPPS[1..RDISPCONNMAX]	NUMEXCRSPMAX
	CDISPAVGPPSCONN[1..RDISPCONNMAX]	NUMFIMINCON
	CDISPAVGSPSCONN[1..RDISPCONNMAX]	NUMFIMOUTCON
	CDISPMAXPPS[1..RDISPCONNMAX]	NUMIOLINCON
	CDISPMAXPPSCONN[1..RDISPCONNMAX]	NUMIOLOUTCON
	CDISPMAXSPSCONN[1..RDISPCONNMAX]	NUMNTFRQUAVG
	CEECOMMAND	NUMNTFRQUMAX
	CEESTATE	NUMPARRSPAVG
	CPEERAVGPPS[1..RPEERCONNMAX]	NUMPARRSPMAX
	CPEERAVGPPSCONN[1..RPEERCONNMAX]	NUMPEERENV
	CPEERAVGSPSCONN[1..RPEERCONNMAX]	NUMSCEINCON
	CPEERMAXPPS[1..RPEERCONNMAX]	NUMSCEOUTCON
	CPEERMAXPPSCONN[1..RPEERCONNMAX]	NUMSIGS
	CPEERMAXSPSCONN[1..RPEERCONNMAX]	NUMSIOLMINCN
	CPUCYCLEAVG[0..39]	NUMSIOLMOUCN
	CPUCYCLEMAX[0..39]	PEERENV
	CPUFREEAVG	PEERGETAVG[1..30]
	CPUFREEMIN	PEERSTRAVG[1..30]
	CRCYCLEOVRN[0..40]	PEERSTRRESP
	CTRLCONFIRM	PEERSUBSCPER
		PRIMARYSIG
		RDISPDEGIMRCONN[1..RDISPCONNMAX]
		REASONSET
		RPEERNAME[1..RPEERCONNMAX]
		SCANASSOCDSP
		SCANCTRLLVL
		SCANGRPDTL
		SCANPNTDTL

	DESC EUDESC EXTGETRQUAVG EXTGETRQUMAX EXTSTRRQUAVG EXTSTRRQUMAX GROUP.NUMPARAMS HIST.NUMPARAMS INALM IPEERCONNERRCODE[1..N] IPEERCONNERRINFO[1..N] IPEERCONNSTS[1..N] IPEERNAME[1..IPEERCONNMAX] IPEERPATH[1..IPEERCONNMAX] LSCYCLEOVRN[0..40] MAXBLKTYPES	SECONDARYSIG SECSIGSECLVL SIMCOMMAND SIMSTATE STATSRESET STRRESP SUBSCPERIOD TREND.NUMPARAMS
Reference	Refer to the *Control Builder Parameter Reference* manual for definitions of each parameter.	

8 Control Functions and Insertion Points

Related topics

“Insertion Point Support” on page 68

8.1 Insertion Point Support

The ACE controller supports CAB insertion points in Data Acquisition and Regulatory Control function blocks. See Control Builder Components Theory

9 Operations

Related topics

“Interacting with ACE” on page 70
“ACE Restarts” on page 73
“ACE Reconfiguration” on page 75
“ACE Checkpoint and Restore” on page 76

9.1 Interacting with ACE

Related topics

"Deleting ACE/CEEACE block" on page 70
"Issuing a shutdown command and using checkpoints to restore ACE" on page 72

9.1.1 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.

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.

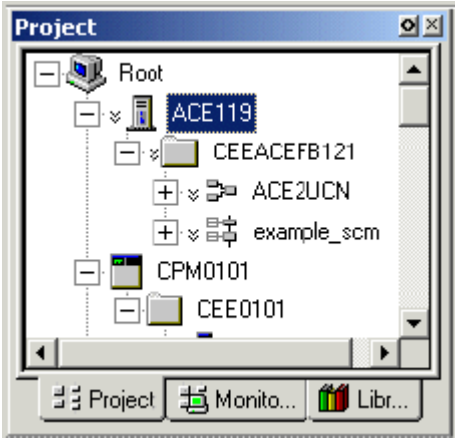
3 Select components contained in CEEACE and click delete button in the tool bar.
Calls up Delete Selected Objects dialog.

4 Click the Deleted Selected Object(s) button.
Initiates the delete function and progress dialog tracks status until complete.

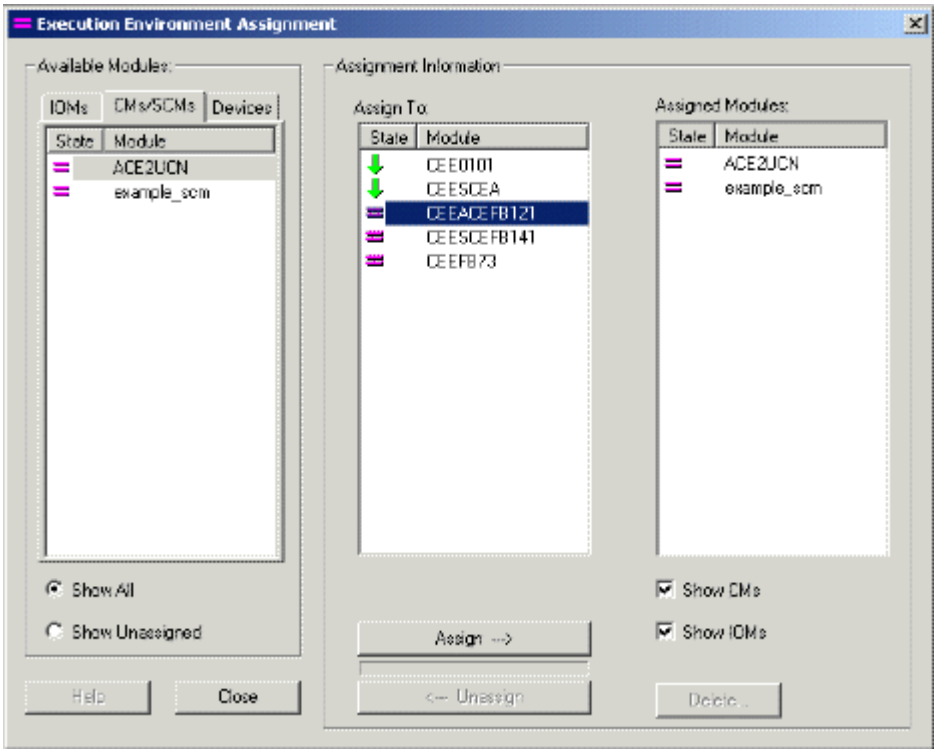
5 Click the ACE block and click the delete button in the tool bar.
Calls up Delete Selected Objects dialog.

6 Click the Delete Selected object(s) button.
Initiates the delete function and progress dialog tracks status until complete.

7 Click the Project tab.
Calls up the Project view.



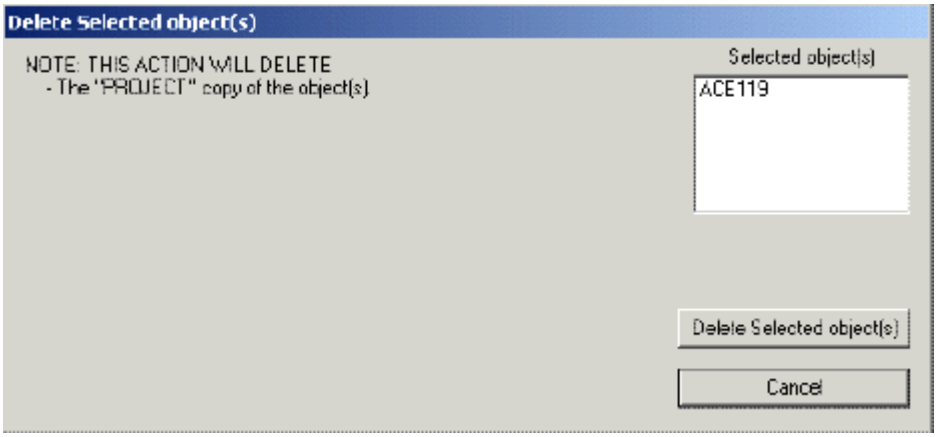
8 With ACE selected, click assign button in the tool bar.
Opens the Execution Environment Assignment dialog box.



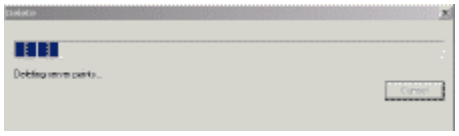
9 Click module assigned to ACE in Available Modules list, select CEEACE in Assign To list, select all the modules listed in the Assigned Modules list and click the Unassign button.
Unassigns modules from CEEACE.

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.



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.

9.1.2 Issuing a shutdown command and using checkpoints to restore ACE

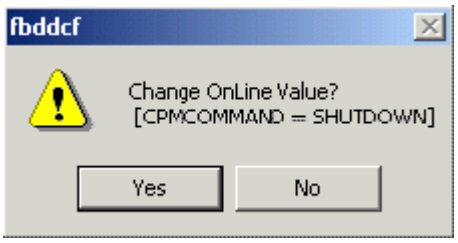
Use the following procedure as a general guide for issuing a Shutdown command to a loaded ACE/CEEACE block through the Monitor mode in Control Builder. This procedure assumes that Control Builder is running with tree views open. The illustrations used in this procedure are for example purposes only.

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 **Checkpoint** > **Restore** from Checkpoint from shortcut menu.
Opens checkpoint restore dialog.

7 At this point, perform the operations described in Restoring from checkpoint

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 snapshot recovery procedure.

9.2 ACE Restarts

Related topics

“Brief review of startup/restart concepts” on page 73
“ACE cold/warm startup” on page 73
“CEE restart option per SCM/CM” on page 74

9.2.1 Brief review of startup/restart concepts

A CEE can undergo different kinds of start up. The most basic is when a CEE is started by initial load of its parent controller / device block. When this happens the CEE starts out empty of algorithm blocks and is basically inert. This sort of restart does not pose any special issues, and it will not be described here.

This section deals with startups in which there is previous CEE data that must be handled correctly. These start ups are usually referred to as "restarts

There are two kinds of restarts discussed in this document:

Restart Type	Characteristics
Cold Restart	This kind of restart always happens as a transition in the value of CEESTATE from idle to run. It can happen after the CEEACE FB has just been created and the initial configuration loaded. A more critical form of cold restart happens after restore of a checkpoint so that the CEE database is neither null nor newly loaded from CB. In this case, a careful policy must be followed as to which data is preserved and which is reinitialized. The policy followed by cold restart assumes that much of the state data is stale. Thus, any live data that can be derived directly from the process is wiped out. But in addition, much of the operational data that was captured in the saved checkpoint is reinitialized. All configuration data is of course saved across cold restart
Warm Restart	Warm restart is directly analogous to cold restart in that it always happens as a transition in the value of CEESTATE from idle to run and in that its most critical form occurs after restore of a checkpoint. It differs in the policy it applies in choosing data to preserve and data to reinitialize. Warm restart preserves all data that cold restart preserves but preserves additional operational data as well.

9.2.2 ACE cold/warm startup

When changing the CEE state from Idle to Run, the operator is given the choice of what type of Restart to be performed. The different types of Restart are characterized as follows.

Restart Type	Behavior
Activate-Cold	UCNOUT and HiwayOut blocks connected to xPM points, and TPS-HG points go into initialization. If supported, any remote cascade requests in the secondary blocks are turned off, requiring the Operator to manually change the modes of all secondary Regulatory Control blocks
Activate-Warm	On the first execution, UCNOUT and HiwayOut blocks connected to xPM, or TPS-HG points go into initialization. Any remote cascade requests from the secondary blocks are responded to, thus setting the secondary block Mode to Cascade. On the second execution, automatic control is resumed

Tip

Regulatory Control blocks connected to C200/FIM blocks are considered peer connections and not supervisory connections. On ACE restart these blocks are only affected by the status of their secondaries. Initialization is driven up the chain from the secondary connected to the I/O and not by the type of ACE restart

9.2.3 CEE restart option per SCM/CM

The ACE node shall support a cold/warm activation option for SCMs and CMs. The options permit the SCM/CM to always start warm, always start cold or always follow the CEE. The options are summarized below:

Restart Option	Behavior of the SCM
FollowCEE	For this configuration parent control modules instruct their component blocks to initialize according to whichever type of restart the operator has selected for the CEE as a whole: cold or warm.
AlwaysCold	The operator must intervene to activate the SCM.
AlwaysWarm	The SCM shall start back at the first transition in the main handler.

9.3 ACE Reconfiguration

Related topics

“Reconfiguring ACE to a computer other than that originally configured” on page 75

9.3.1 Reconfiguring ACE to a computer other than that originally configured

The user can quickly reconfigure an ACE to another computer to correct configuration errors or in case of a computer problem. The overall procedure:

1. “Deleting ACE/CEEACE block” on page 70.
2. “Creating an ACE and CEE” on page 52.
3. Reload the ACE.
4. Reload all peer strategies which reference the reconfigured ACE.

9.4 ACE Checkpoint and Restore

Related topics

“Checkpoint and restore operations” on page 76

9.4.1 Checkpoint and restore operations

Checkpoint and restore operations are covered in the *Control Building Guide*.

10 System Administration

Related topics

“Local ACE Node Administration” on page 78

10.1 Local ACE Node Administration

ACE node administration is done locally. Keyboard, mouse, and monitor should be available to connect to the ACE node, when reviewing/adjusting administration settings. Node administration function includes:

- checking status of node services,
- using node performance tools, or
- starting and stopping other applications on the node

Attention

The system does not prevent the user from installing and executing other applications on the same node as ACE or Experion applications. For robust and reliable ACE control, it is highly recommended that only ACE related applications be hosted on the ACE node.

11 Troubleshooting and Maintenance

Related topics

11.1 Recovering from ACE node 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.

Prerequisites

The following state should exist, representing a fully functional, active ACE node:

- ACE node is running
- CEEACE function blocks are loaded
- CM/SCM with strategies built are loaded and active

Indications of failure

If the ACE loses power the following occurs:

- Loss of Communications with Controller event is generated
- Errors are indicated on the Monitoring tab icons for the ACE, CEEACE, and CM/SCM FBs associated with ACE
- Lower level strategies dependent on the ACE for supervisory control sheds to their configured backup modes.

11.1.1 Recovering from node power failure

Use the following procedure to recover from an ACE power failure

1. Restore power to ACE node.
   Wait for ACE node to return to operation (Boot).
2. Right click ACE icon and select **Checkpoint** > **Restore** from Checkpoint from shortcut menu.
   Opens checkpoint restore dialog.
3. At this point, perform the operations described in Restoring from checkpoint
4. 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.
5. Click the Yes button to continue.
   Activates components and icons turn green.
6. This completes the power failure recovery procedure.

11.2 Recovering from ACE application failure

Use the following procedure as a general guide for recovering ACE/CEEACE operation after an ACE application failure. This procedure assumes that Control Builder is running with tree views open. The illustrations used in this procedure are for example purposes only.

Prerequisites

The following state should exist, representing a fully functional, active ACE node:

- ACE node is running
- ACE/CEEACE function blocks are loaded
- CM/SCM with strategies built are loaded and active

Indications of failure

An ACE application running

An ACE application failure occurs when the either the CDA-sp or EE component or both fail. If the ACE application fails the following occurs:

- Loss of Communications with Controller event is generated
- Errors are indicated on the Monitoring tab icons for the ACE, CEEACE, and CM/SCM FBs associated with ACE

11.2.1 Recovering from application failure

Use the following procedure to recover from an ACE application failure:

1. Restore power to ACE node.
   Wait for ACE node to return to operation (Boot).
2. Right click ACE icon and select **Checkpoint** > **Restore** from Checkpoint from shortcut menu.
   Opens checkpoint restore dialog.
3. At this point, perform the operations described in Restoring from checkpoint
4. 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.
5. Click the Yes button to continue.
   Activates components and icons turn green.
6. This completes the recovery procedure.

11.3 Recovering from an I/O connection error 2299

Prerequisites

The following state should exist, representing a fully functional, active ACE node:

- ACE node is running
- ACE/CEEACE function blocks are loaded
- CM/SCM with strategies built are loaded and active

Indications of failure

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.

11.4 Recovering from OPC Server failure

Prerequisites

The following state should exist, representing a fully functional, active ACE node:

- ACE node is running
- ACE/CEEACE function blocks are loaded
- CM/SCM with strategies built are loaded and activè
- OPC Server function block is successfully loaded
- ACE contains control strategies that reference the OPC Server
- Communication between the ACE and the OPC Server is normal

Indications of failure

An OPC Server failure happens if any of the following occurs:

- OPC Server component fails
- The single Ethernet connection to the OPC Server fails
- OPC Server loses power

If OPC Server fails the following occurs:

- Communication errors from ACE strategies that reference OPC Server data. Several process alarms can be reported as a result of:
  - fail-safe data substitution for Gets, and
  - failures for Stores

11.4.1 Recovering from OPC Server failure

Use the following procedure to recover from an OPC Server failure:

1. Restore power to the OPC server node, if power was lost.
2. Repair the failure in the communication link, if needed.

11.5 Resolving Overruns or Crashes in ACE Running Instances of CAB Types

Diagnostic check

An ACE running instances of CAB types either gets overruns or crashes.

Cause

Instance of CAB types that have string or time Value Custom Data Parameters (CDPs) that are wired for input and are either distributed CAB types or an atomic or distributed CAB type that goes to a DORMANT state can cause the memory to increase.

Solution

Use Parameter References for inputs of string and time data types instead of CDPs, so you can control when data is retrieved from the remote blocks.

11.6 Preventing the CDA-sp crash during peer-to-peer communication between ACE and another controller through Inter Cluster Gateway (ICG)

Diagnostic check

Peer-to-peer communication between ACE and another controller (ACE, C200, C300, and so on) communicating through the ICG does not work.

Cause

When performing a 'store' from one ACE to another controller using the ICG, there may be continuous disconnection/reconnection events with the target controller. This may result in the Experion PKS CDA-sp service to be terminated on the machine hosting the ICG.

Solution

When performing a 'store' from one ACE to another controller through the ICG, you must configure at least one 'READ' Control Module between the ACE and the other controller.

For example, if ACE1 is 'storing' to a C300 using ICG then configure a Control Module on ACE1 such that it reads some parameter from the C300 through the same ICG. The parameter being read could be any parameter from the C300 or its blocks.

If you do not configure this CM, the CDA-sp service hosting the ICG may terminate unexpectedly.

12 Notices

Trademarks

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

OneWireless™ is a trademark of Honeywell International, Inc.

Other trademarks

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

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

Third-party licenses

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

12.1 Documentation feedback

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

http://www.honeywellprocess.com/support

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

hpsdocs@honeywell.com

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

12.2 How to report a security vulnerability

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

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

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

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

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

- Send an email to security@honeywell.com.

  or

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

12.3 Support

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

12.4 Training classes

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