Honeywell

Experion PKS
CL Server User's Guide

EPDOC-XX12-en-431A
February 2015

Release 431

Honeywell

Document	Release	Issue	Date
EPDOC-XX12-en-431A	431	0	February 2015

Disclaimer

Contents

1 About this document

This document describes how to use the CL Server on the network from a server perspective. It does not cover the CL Application initiation from the TPN (LCN). The primary user tasks are installation, configuration, and operation (startup and shutdown) of the CL Server. Additionally, you will access and view the System Management display for CL Server.

References

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

Document Name	Document ID
System Management Operations Guide	EPDOC-X141-en-A
System Management Configuration Guide	EPDOC-X142-en-A
Configuration Utility User's Guide	EPDOC-XX14-en-A
OPC Specification Reference Manual	TP41230

2 Introduction

Related topics

“Introduction to the CL Server” on page 8
“Requirements for using the CL Server” on page 9

2.1 Introduction to the CL Server

This guide describes how to use the CL Server on the TPS Network (TPN) from the server perspective. It does not cover the CL Application initiation from the TPN (LCN).

The primary user tasks are installation, configuration, and operation (startup and shutdown) of the CL Server. Additionally, you will access and view the System Management display for CL Server.

2.1.1 Features

The CL Server User Guide provides information for you to do the following.

- Verify the CL Server installation.
- Configure the CL Server.
- Operate the CL Server (startup, viewing status displays, and shutdown).

The CL Server accepts requests from the TPN to initiate, run, and terminate Windows 2008-side applications. It retains all the functionality of previous CL controllers. Any CL-initiated applications that ran on an Application Module Personality (AMW) will run on a TPS system.

The CL Server is an HCI managed component, and can be viewed and manipulated from the System Status Display.

A windowless application on the Windows 2008-side can be started and stopped by a CL program on the TPN side at anytime. CL Initiated Applications are also managed using the System Management Display. The CL Server component Status Display will show the status of the CL Server and initiated applications.

2.2 Requirements for using the CL Server

2.2.1 User skill prerequisites

This document is a guide for navigating and accessing information that will allow the CL Server to become functional. It is assumed that the user is skilled in performing Windows administration tasks and has a working knowledge of the Experion system.

Attention

CL Server implementation requires that you are logged on as both a local administrator to install server component software. When in a domain environment, to configure the CL Server you must be logged on as a Product Administrator as to configure the HCI component and configure security. Therefore, you must be familiar with the system policies contained in the *Windows Domain and Workgroups Implementation Guide*.

2.2.2 System environment

The CL Server may run on a Windows 2008 server optionally configured as part of the TPS domain. This workstation must be directly connected to the TPN through an LCNP board in the PC. The network requires the configuration for use of DCOM.

2.2.3 Network environments

Item	Requirement
Network	• The network requires the availability of a Windows Domain Controller (Windows Server 2008). • If using workgroups, domain functions TPS Domain cannot be introduced.
PIN/PCN	Ethernet, TCP/IP, Microsoft application layer software such as RPC & DCOM. One of the following name services must be available: • DNS - Required for Windows domains • WINS - Optional and recommended

2.2.4 How to use this guide

This guide is organized into the following major sections that represent user tasks:

The *Verifying the CL Server Installation* section provides a method of verifying the installation of the CL Server component software. This section may be skipped if the CL Server is already installed and functional.

The *Configuring the CL Server* section describes how to configure the CL Server as a component in an optional TPS Domain.

The *Monitoring CL Server Status* section provides operational procedures that allow you to view status, start, and stop the CL Server.

The *Supporting CL Application's* section provides information about CL-initiated Windows applications.

3 Verifying the CL Server Installation

3.1 Intended users

This section is of interest primarily to the system administrators or to anyone who needs to verify the installation of the CL Server software package on their system. If CL Server is already installed and functioning on your system, you may skip this section.

Attention
The intent of this section is to provide a method of verifying the installation of the CL Server software. This includes the CL Server software package according to the licensing agreement. Please consult that publication for installation-related tasks.

3.1.1 What the CL Server package represents

The CL Server is a specific package component within the Experion software package.

The CL Server software package must be installed on a Experion APP node that contains an installed LCNP board. Currently, there can be only one CL Server for each Experion APP Node.

The CL Server software installation can be verified assuming the Experion APP Base package is installed.

Attention
Verify that the CL Server software is installed on an **Experion APP Node**. The node that is assigned to run the CL Server must contain an LCNP board. Currently, there can be only one CL Server per LCNP board.

For more information about Experion software installation, refer to the *Experion Software Installation User's Guide*.

3.1.2 Verification of a successful installation

Upon successful CL Server installation, your selected drive directory should include a folder called CLAppServer containing the contents similar to the items shown in the following figure.

Attention
The contents of the CLAppServer folder will vary depending on what kind of node is being used and what software packages are being installed.

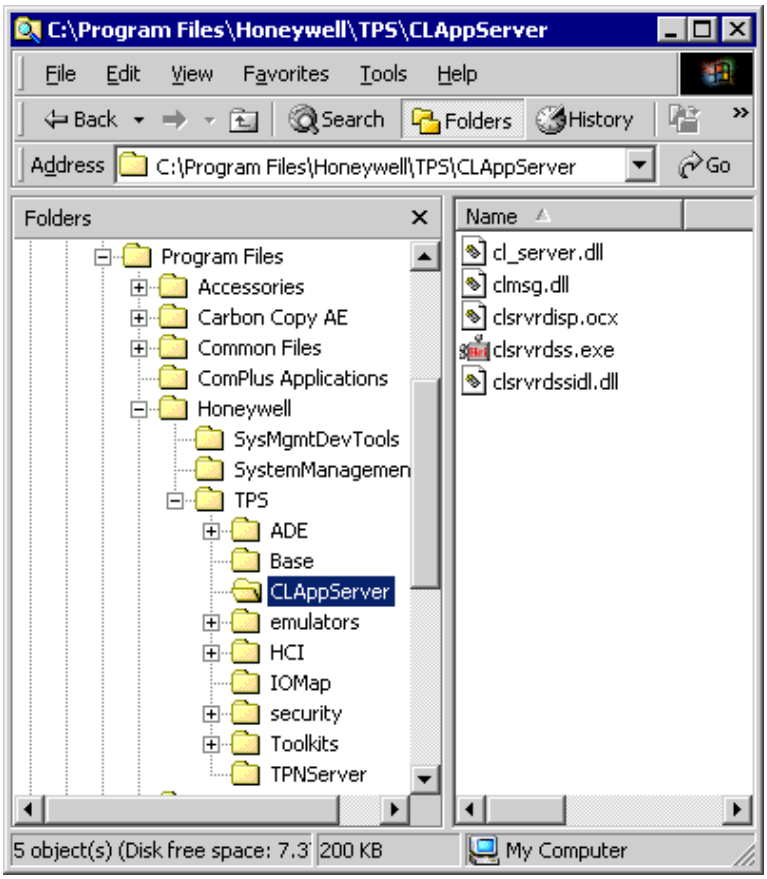


Figure 1: Directory with the CLAppServer Folder

4 Configuring the CL Server

4.1 Approach to CL Server Configuration

This section provides the following information about CL Server configuration.

- CL Server pre-configuration requirements
- CL Server configuration
- CL Server installation test
- CL Server configuration removal

4.2 CL Server Pre-configuration Requirements

The following table lists CL Server pre-configuration requirements.

Pre-configuration Requirement	Note the Following	For Additional Information
Configure the Application Processing Platform (Experion APP) node	Be sure to select the CL Server when installing Experion Software. The *CL Server* should be installed on the Experion APP Node where the component will run.	Experion Software Installation Guide
Install the connectivity client on nodes that access the CL Server	The *Experion APP TPS Client* should be installed on nodes from which the component will be configured or monitored. Note that the *Experion APP TPS Client* item is automatically selected when the GUS HCI Client Add-In package is selected.	Experion Software Installation Guide

4.2.1 Additional pre-configuration consideration

The node hosting the CL Server establishes its HCI Name Service scope based on the hosting node's environment.

- If the node is in a workgroup, its HCI Name Service scope is the workgroup.
- If the node is in a domain, its HCI Name Service scope is the domain.
- If the node is in a TPS Domain, its HCI Name Service scope is the TPS Domain.

The HCI Name Service is used by HCI client applications to resolve an alias name to a server's CLSID and computer name. The alias name for CL Server must be unique within its scope.

For more information about System Management facilities such as HCI Name Service, refer to the *System Management Configuration Guide*.

4.3 CL Server Configuration

CL Server configuration consists of defining the server as an HCI managed component. The following table provides an overview of how to configure a CL Server.

1 Verify the pre-configuration requirements listed previously are in place.
   Experion APP node and client nodes have base components installed.
2 Access the HCI Component Configuration page using either the System Management Display or Configuration Utility.
   The HCI Component Configuration page provides definable CL server properties.
3 Configure the CL Server using the component properties to define the following:
   - Select HCI CL Server as the BASE Prog ID property.
   - Name the CL Server (for example: CL Server1) using the Component Name property.
   - Determine security (for example: access levels) using the Secured Methods and Capability properties.
4 A CL Server is configured and ready to support client application access.

4.3.1 Enabling CL client applications started using command prompt

It is observed that on Windows 2008 Server 64-bit based EAPP nodes, CL client applications (that are developed for 16-bit/32-bit operating systems) do not work if they are started from command prompt.

Attention
With this configuration, CL server functionality will not be available if you logoff from EAPP.

To enable CL client applications if they are started from command prompt, configure CL server to run as an interactive user in the DCOM settings by performing the following:

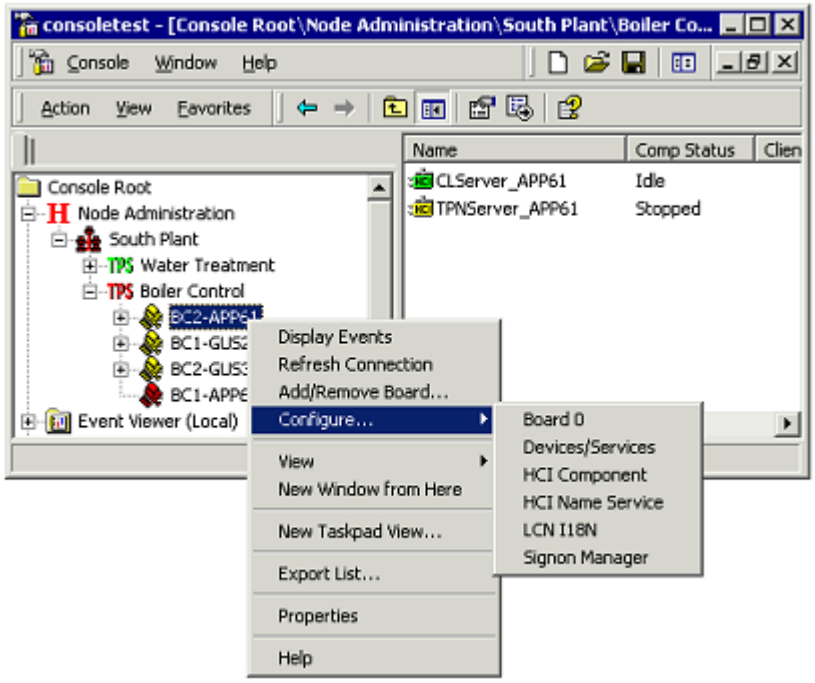
1 Type **dcomcnfg** in the Run command.
2 From the console, browse to **Console Root** > **Component Services** > **Computers** > **My Computer** > **DCOM Config**.
3 Search for **HCI_CL Server - exe Server**, right click and then select **Properties** to modify the settings.
4 From the Identity tab, select **The interactive user** option and apply the changes.
5 Restart the node or logoff/login to start CL client applications from command line or directly from the CL server.

4.4 Invoking HCI Component Configuration Page

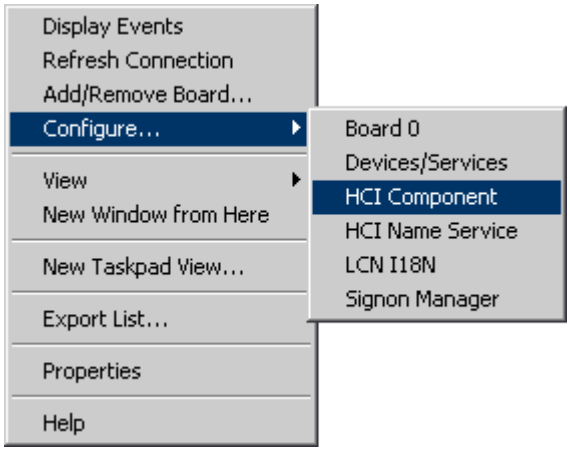
HCI Component Configuration operates within the System Management Display that is implemented as a Microsoft Management Console (MMC) snap-in. Alternatively, the HCI Component page is also accessible from the Configuration Utility. To configure a CL Server, first access the HCI Component Configuration page using one of the following procedures.

Access HCI Component Configuration using System Management Display

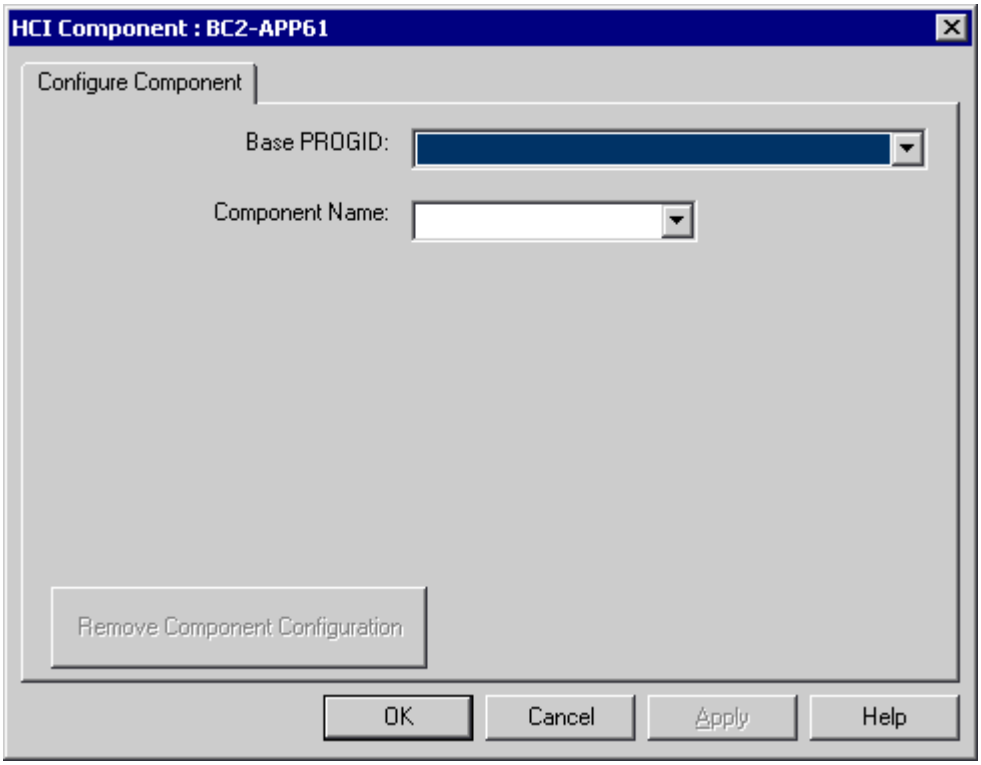
1 Right-click the Experion APP node from the System Management Display hierarchy and select **Configure.**



2 Select **HCI Component** from the menu to begin CL Server configuration.



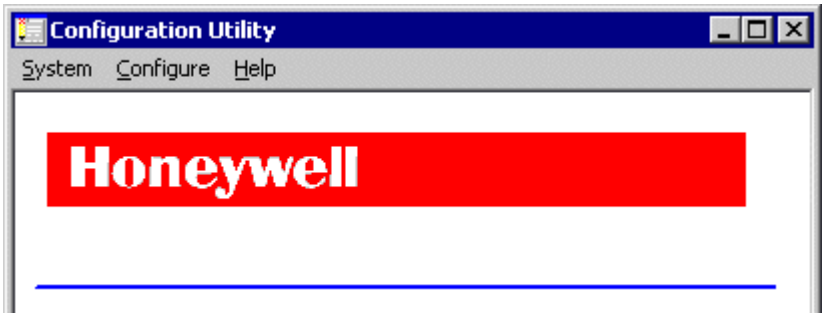
The HCI Component dialog appears.



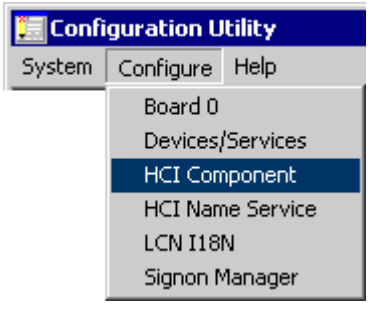
Access HCI Component Configuration using Configuration Utility

1 Select **Start Menu** > **All Programs** > **Honeywell Experion PKS** > **System Management** > **Configuration Utility**.

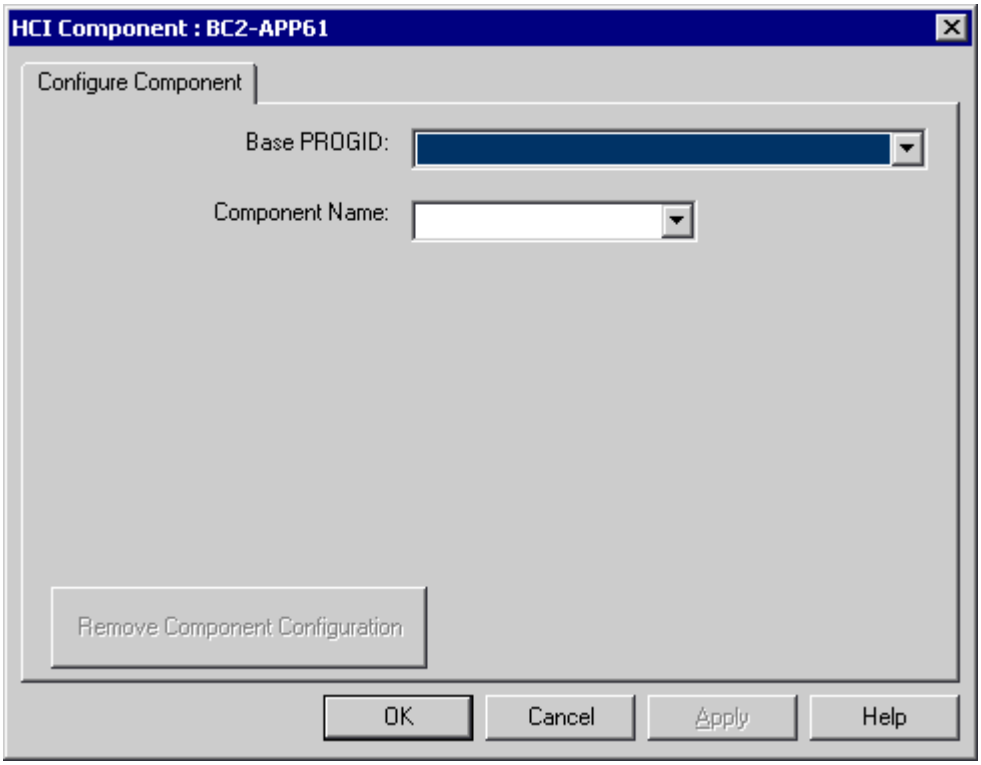
The Configuration Utility appears.



2 Select **Configure** > **HCI Component**.



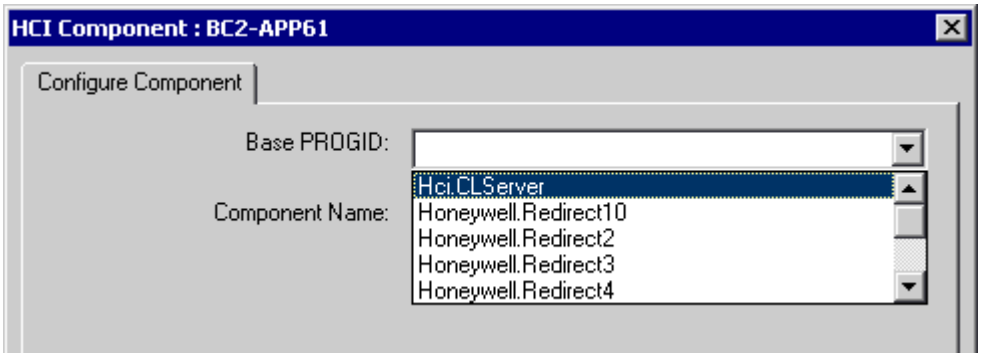
The HCI Component page appears.


HCI Component : BC2-APP61
Configure Component
Base PROGID:
Component Name:
Remove Component Configuration
OK
Cancel
Apply
Help

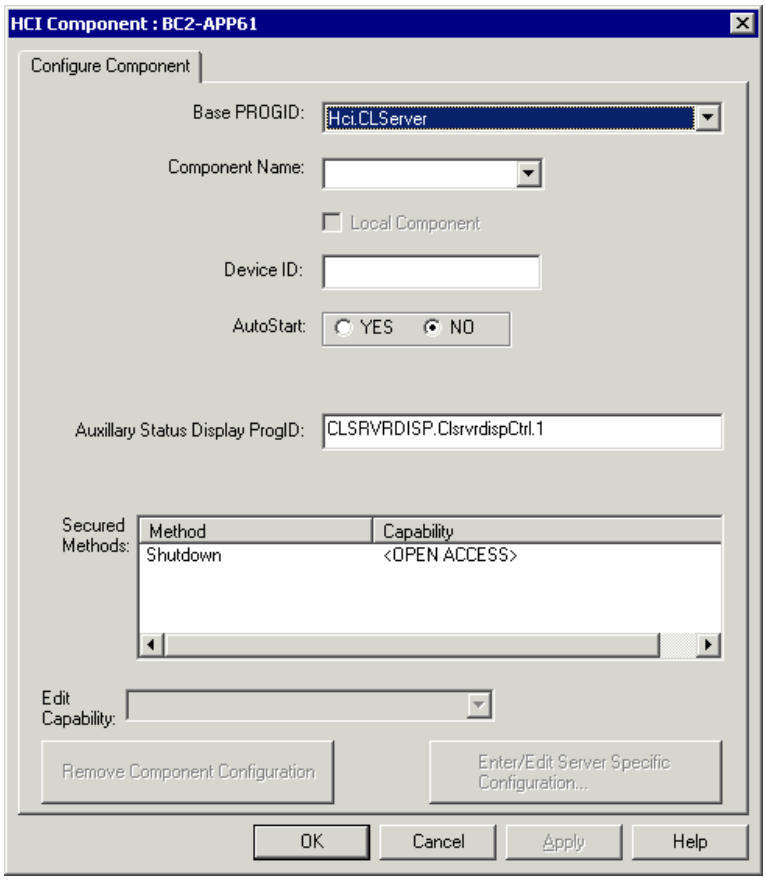
4.5 Configuring CL Server from HCI Component Page

Procedure to configure CL Server

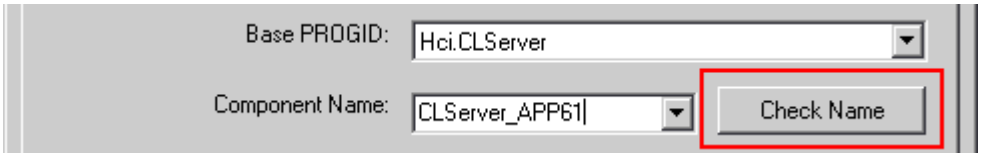
1 Select **Hci.CLServer** from the **Base PROGID** drop-down list.
The Configuration Page displays definable CL Server properties.



2 Enter a name in the **Component Name** field for the CL Server as per the requirement.



3 Click the **Check Name** button to verify the validity of the Component Name.



4 Select the **NO AutoStart** option if the CL Server is to be manually started
Or,

Select the **YES AutoStart** option if the CL Server is to be started at every system restart.

5 Click the **Enter/Edit Server Specific Configuration** button.

6 Click the **OK**button for each message box that appears until just the CL Server Security Configuration page is displayed and then continue to the *Selecting CL Server Security*section.

4.6 Selecting CL Server Security

Example Server Security Configuration page

1 An example of the **CL Server Security Configuration** page is shown in the following figure.

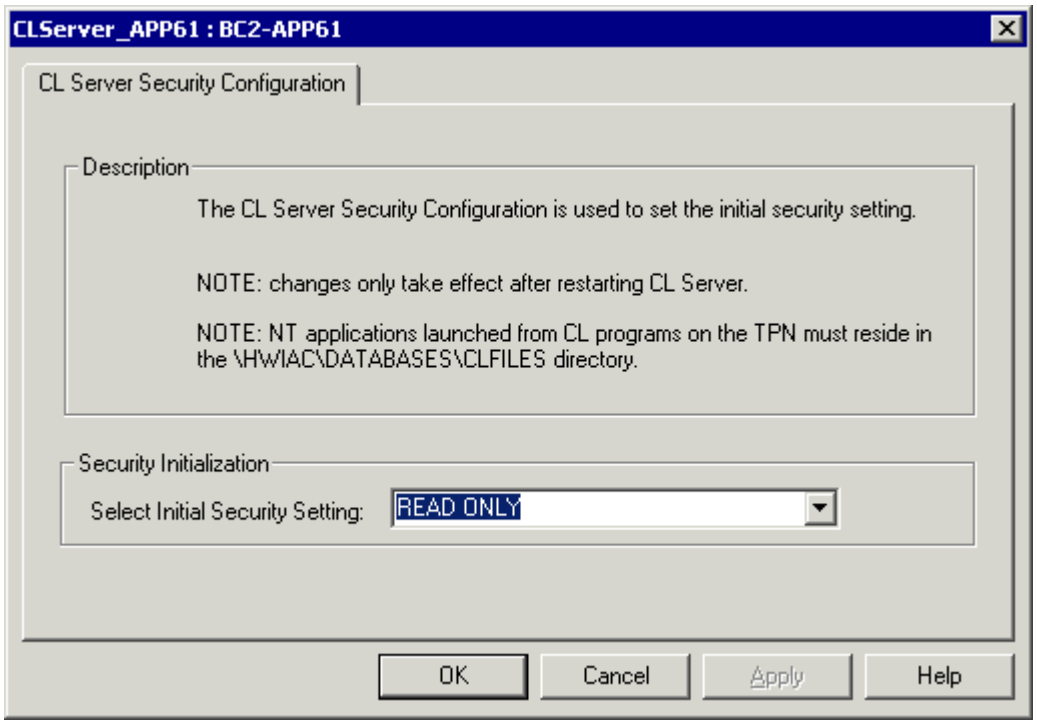


Figure 2: CL Server Security Configuration Page

2 The global X-access switch controls when a Windows 2008-side application can write TPS Network data. The **Initial Security Setting** shown on the **CL Server Security Configuration** page provides the startup value for the global X-access switch.

3 It should be noted that this switch is only used when the CL Server is running. If the CL Server is not running, all Windows 2008-side applications will be able to write TPS Network data. The switch is implemented by a Processor Status Data Point (PSDP) parameter called $XACCESS which has three possible states:

- READ ONLY-The Windows 2008-side can read but cannot write TPS Network data (default).
- READ WRITE ONLY FOR CL INITIATED-The Windows 2008-side can read TPS Network data, but can write TPS Network data only from applications that are initiated by CL on the TPS Network-side.
- READ WRITE-The Windows 2008-side can read and write TPS Network data from CL-initiated and non-CL-initiated applications (this state cannot be set by any means unless the external load module XACCESS is loaded).

Attention

For nodes that run CL Server and TPN Server the **Initial Security Setting (**$XACCESS) must be set to READ-WRITE. Currently, $XACCESS security is limited when Windows 2008 Applications use the TPN Server to access TPN data. The TPN Server only supports read-only access and read-write access. When read-write only for CL Initiated Applications is selected, the TPN Server will default to read-only access.

The setting configuration is described in the following procedure.

1 Select a security setting from the **Select Initial Security Setting** pull-down menu using the preceding security setting description information.

2 Click **OK** to re-display the HCI component Configuration page.

4.7 CL Server Configuration Test

1. To access the System Management Display, choose **Start Menu** > **Programs** > **Honeywell Experion PKS** > **System Management** > **System Management Display**.
2. If the **AutoStart** option for the CL Server is selected, then the System Management Display should show that the CL Server has started

   Or,

   If the **AutoStart** option is not selected, then start the server by right-clicking the server name in the System Management Display and then click **All Tasks** > **Start**. Click **Yes** to confirm the start.

4.8 CL Server in a Workgroup

During installation of CL Server, the install package attempts to determine if CL Server is being installed in a domain or workgroup environment. If installed in a domain, CL Server uses a TPS domain account. If installed in a workgroup, CL Server uses a LocalComServer local account. The respective accounts are available from the workstation security template.

Attention

The DCOM configuration setting must be changed If you face any technical problem in configuring/using the CL Server.

If it is necessary to change the DCOM configuration to use a local account from a domain account, you can change accounts using dcomcnfg.

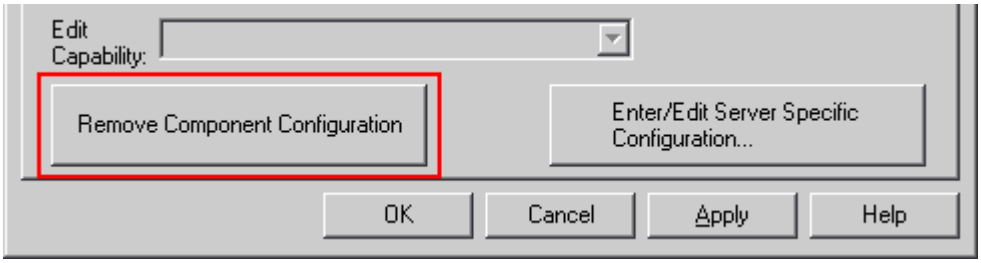
- Choose StartMenu and type dcomcnfg in the command prompt.
- Select Hci.CLServer.
- From the Identity tab, change the user to 'LocalComServer'.
- Enter a password for this account.
  - If you do not know the accounts' password you will have to change it to something you know.
  - If you change the password, you must change it on all nodes in the workgroup.
  - After changing the password, you will need to resynchronize all Windows Services and DCOM servers that use the LocalComServer account to the new password. This also must be done on all nodes in the workgroup using the password utility, **Windows Services & DCOM Servers Log on tool**.

Refer to 'Password Security and Synchronization' in the *Integrated Experion-TPS User's Guide* for procedures on how to use the password utility.

4.9 CL Server Configuration Removal

Procedure to remove component configuration

1. Right-click the node that contains the CL Server from the System Management Display.
2. Select **Configure** > **HCI Component** from the Context menu.
   The HCI Component Configuration Page appears.
3. Select CL Server from the **Component Name** list to display the **Configure Component** page.
4. Click the **Remove Component Configuration** button.



5. Access the HCI Name Service to verify the name is removed from the HCI Name Service Repository.
   - Right-click the node that contains the CL Server.
   - Select **Configure** > **HCI Name Service**.
6. From the HCI Name Service page, select **Refresh List** and verify the name is removed.

5 Monitoring CL Server Status

5.1 Overview of CL Server Status Operations

Because the CL Server is a managed component in the Experion system, it appears in the System Management Display. The CL Server provides status information to both the scope pane and the results pane of the System Management Display.

The managed CL Server can be configured to be auto-started upon system power-up by the Component Administration Service (CAS) provider. Alternatively, a managed component may be manually started from the System Management Display or from a client with appropriate access levels.

This section describes how you can manually start up and shut down a CL Server through the System Management Display.

5.1.1 Typical user interface operations

From the System Management Display, you can:

- Start up the CL Server
- View the status of HCI Components including the CL Server
- Invoke the CL Server Auxiliary Status Display
- Shut down the CL Server

The information in the following procedure is basic reference material for the System Management Display. The procedure provides a description of the display and defines the entities used within the status display window.

5.2 Interpreting System Status Indicators

The following shows the System Management Display with the scope and results pane. The scope pane shows the Node Administration snap-in as the root of the tree and each of the static computers, domains, or TPS Domains selected for monitoring. The results pane shows the status of an HCI managed component.

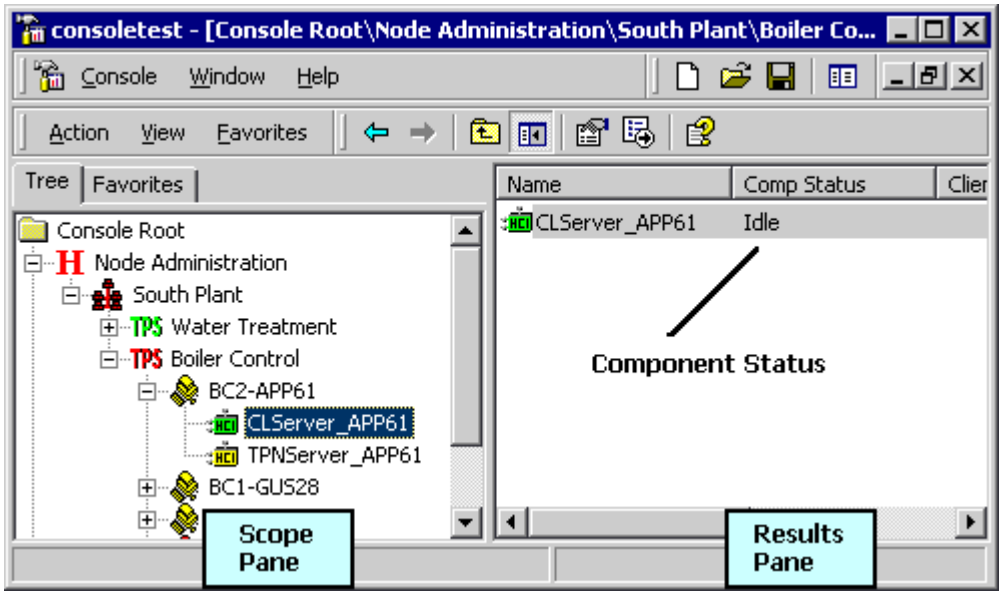


Figure 3: System Management Display

5.2.1 Scope pane

The scope pane shows a tree display composed of several levels. Double-click a level to expand the tree display until you can view the HCI Managed Component that is of interest.

The scope pane (left pane) provides an icon that gives you a quick visual indication of the state of the component. The icon can be one of three colors as shown in the table below:

Component State Color (Scope Pane Icon)	Component State Value
Green	Component is in the **Running** or **Idle** State. The TPS Network Personality is running and the server is fully functioning.
Yellow	Component is in the **Warning** or **Stopped** state. A) It is waiting for the TPS Network Personality to be loaded and transition to the running state on the LCNP in the local node. B) It has lost its connection to the TPS Network Personality.
Red	Component is *not* in **Running** or **Warning** State. The System Management Display will default to the **Stopped** state if the CL Server is not running.

5.2.2 Results pane

The results pane (right pane) presents the status for the selected item from the scope pane. The results pane presents information in configurable columns. The following figure lists properties that can appear in the results pane.

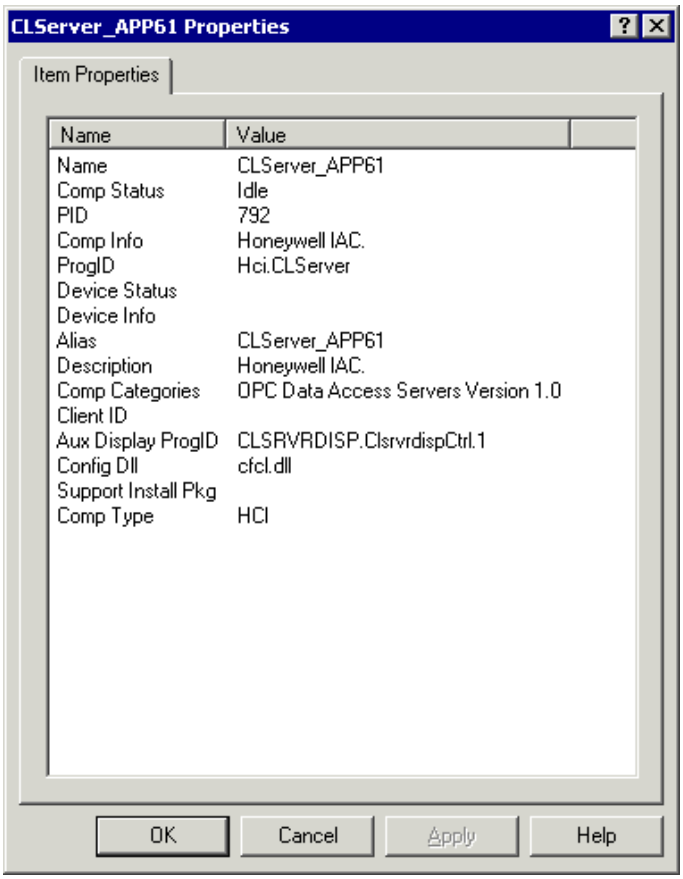


Figure 4: CL Server Properties

To change column width for any of these columns, drag the column separator to the left or right. To sort a column, click on the column header.

The Information column field is a string of 40 characters or less. The CL Server will only provide useful (non-NULL string) information for this field if it is in the Warning State.

5.2.3 Component status

The Results pane also provides a Component Status column that contains the status of the component. The possible status values are:

Components Status	Icon Color	Server Status Values
Stopped	YEL	Component is not currently operational, but has been configured on the node.
Initializing	YEL	Component is initializing.
Test	YEL	Component is in a testing state.
Idle	GRN	Component is operational, and no CL Initiated Applications are running.
Running	GRN	Component is operational, and CL Initiated Applications are running.
Warning	YEL	Component is running, but it has a problem that is not disabling the component.
Shutting down	YEL	Component is shutting down.
Failed	RED	Component terminated abnormally.

5.2.4 Synchronization between Nodes

The System Management Displays on the TPS Network update independently of each other. This means that displays on different nodes may show a different status between configurable update cycles (typically every 15 seconds).

5.3 Starting the System Management Display

Use one of two methods to start the System Management Display.

- Invoke using the Start menu
- Invoke from the Run command

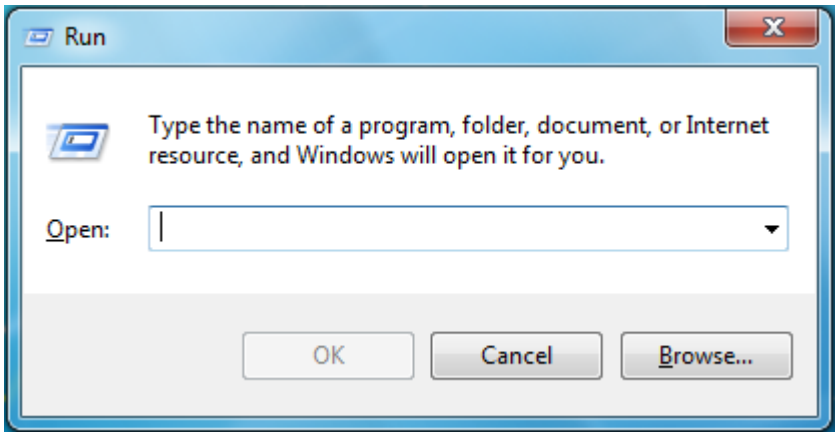
Invoke the System Management Display from the Start menu

- Invoke the **System Management Display** by performing the following:
  Choose **Start Menu** > **Programs** > **Honeywell Experion PKS** > **System Management** > **System Management Display**.

  Click *[console]***.msc.**

Invoke the System Management Display from the Run command

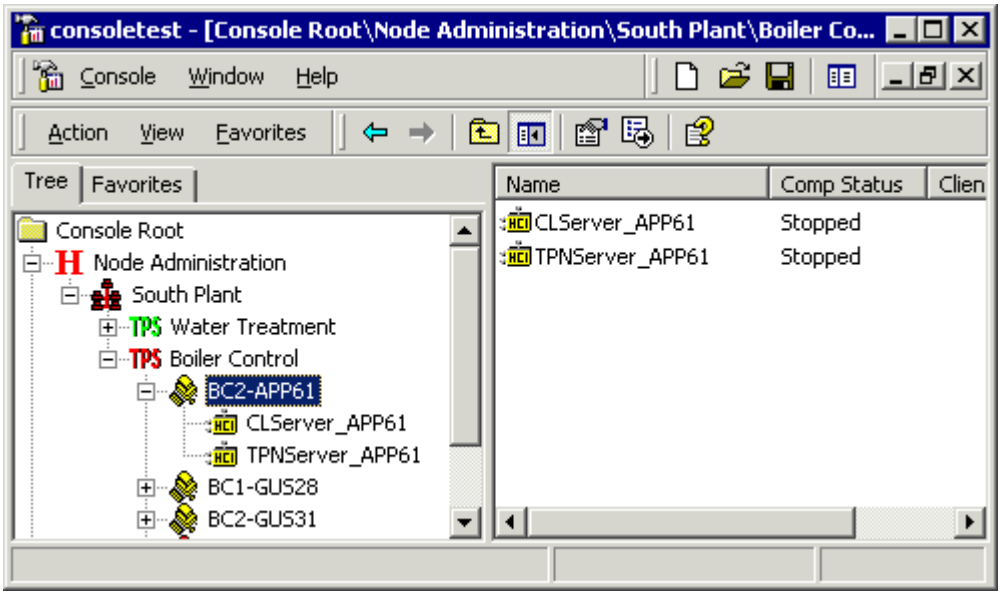
1 Invoke the **System Management Display** by using the **Run** command.
  Type **mmc** in the **Run** dialog.



2 Select a previously saved console view when the MMC appears.

5.4 Starting the CL Server

1. Verify the APP node personality is loaded and running on the LCNP board before starting the CL Server.
2. Invoke the System Management Display using one of the methods in the procedure *Starting the Status Management Display*.
3. The **System Management Display** appears showing icons (in tree form) in the scope pane (left side) and HCI component information in the results pane (right side).



4. **Select** or **click** the node icon where the CL Server is configured (this example uses **CLServer_APP61**).
   Example: The **State** of the HCI component **CLServer_APP61** is indicated as **Stopped**.

   Note: Visual State of Component (icon color)

   Green: Running or Idle state

   Yellow: Warning state (example)

   - Waiting for APP Node Personality to load
   - Lost connection to APP Node Personality

   Red: Not in Running or Warning state

   Component Status

   Idle: Server is operational, and no CL-Initiated Applications are running

   Running: Server is operational, and CL-Initiated Applications are running

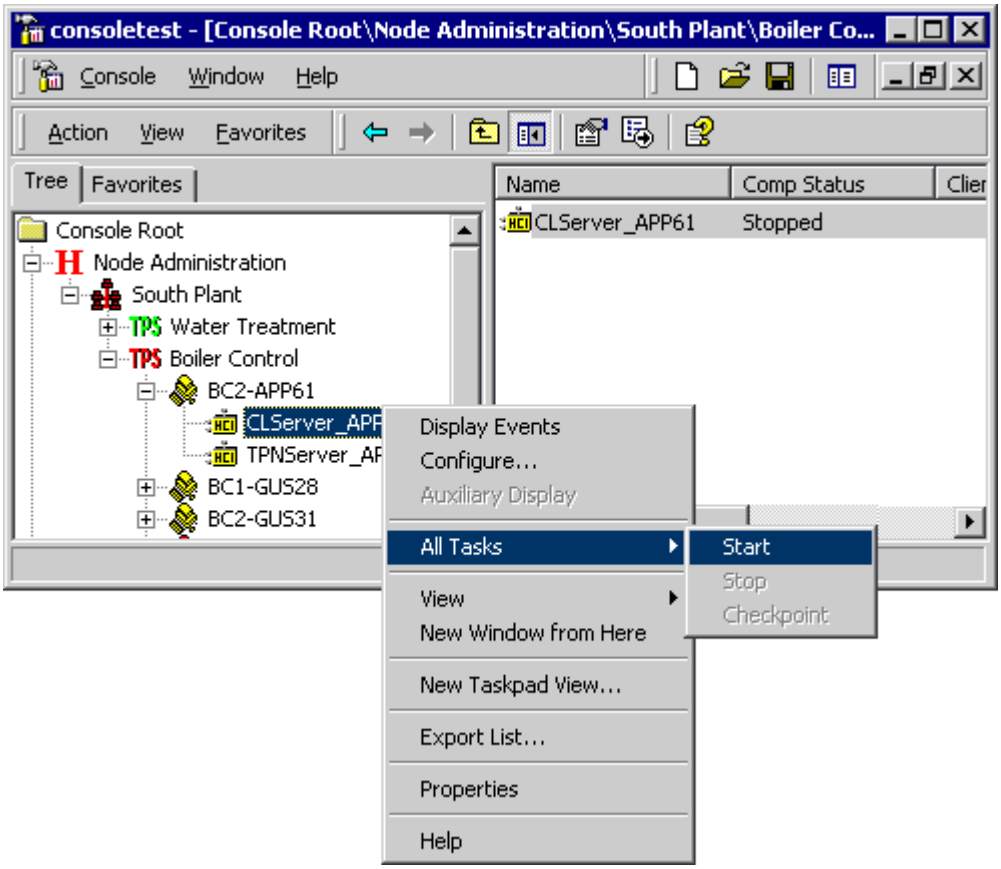
   Stopped: Server is configured, but not running.

   Warning: APP Node personality is not loaded in LCNP or connection to TPN has been lost.

5.4.1 Confirmation of CL Server Startup

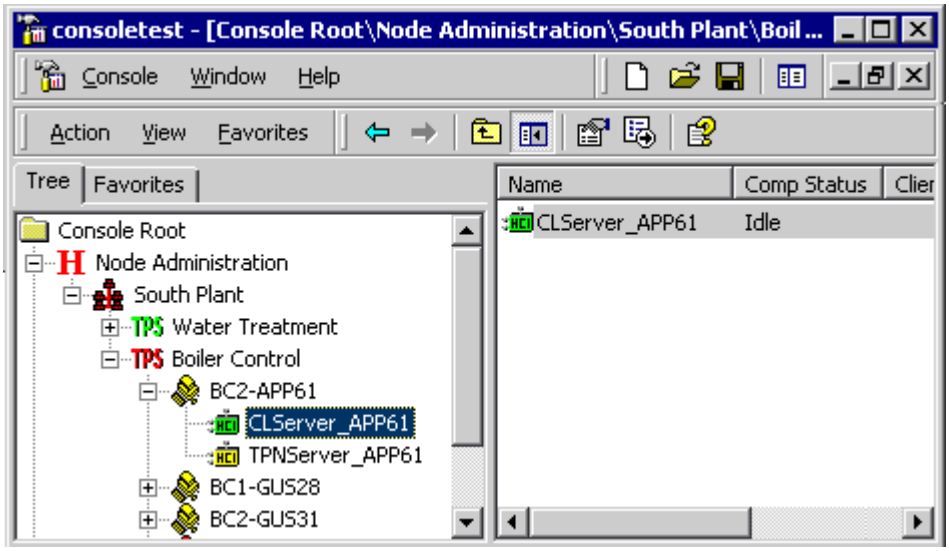
- If the **CL Server** is in the **Stopped** state, it can be started up.
  Right click the HCI component in the results pane (this example uses **CLServer_APP61)**.

  Click **All Tasks** > **Start** from the context menu.



5.4.2 CL Server is Started and in Idle Mode

- Observe the **State** column in the status frame changed from **Stopped** to **Idle**.
  See step 3 for **Status** descriptions.



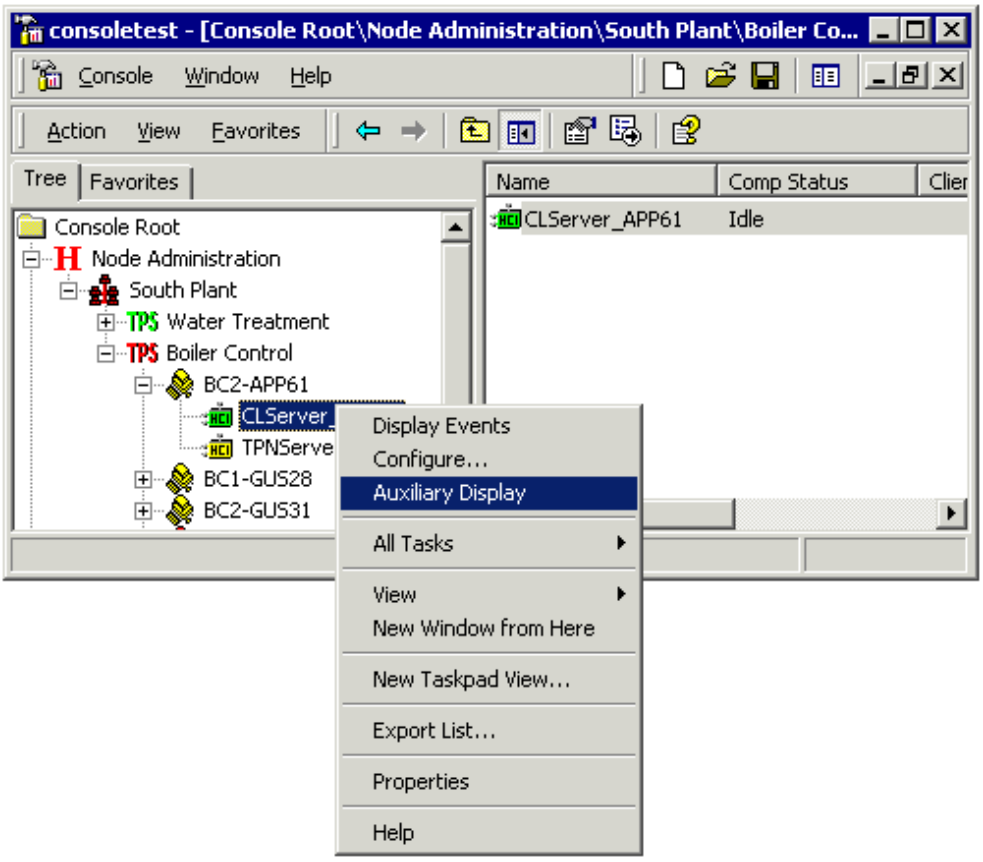
5.5 Invoking the CL Server Auxiliary Status Display

The Auxiliary Status Display provides additional information to engineering personnel and is described in more detail in the following sections.

1 Invoke the System Management Display using one of the methods in the procedure *Starting-up the System Management Display*.

2 **Right click** the selected CL server. This example uses **CLServer_APP61**.
A pop-up menu appears with several entries.

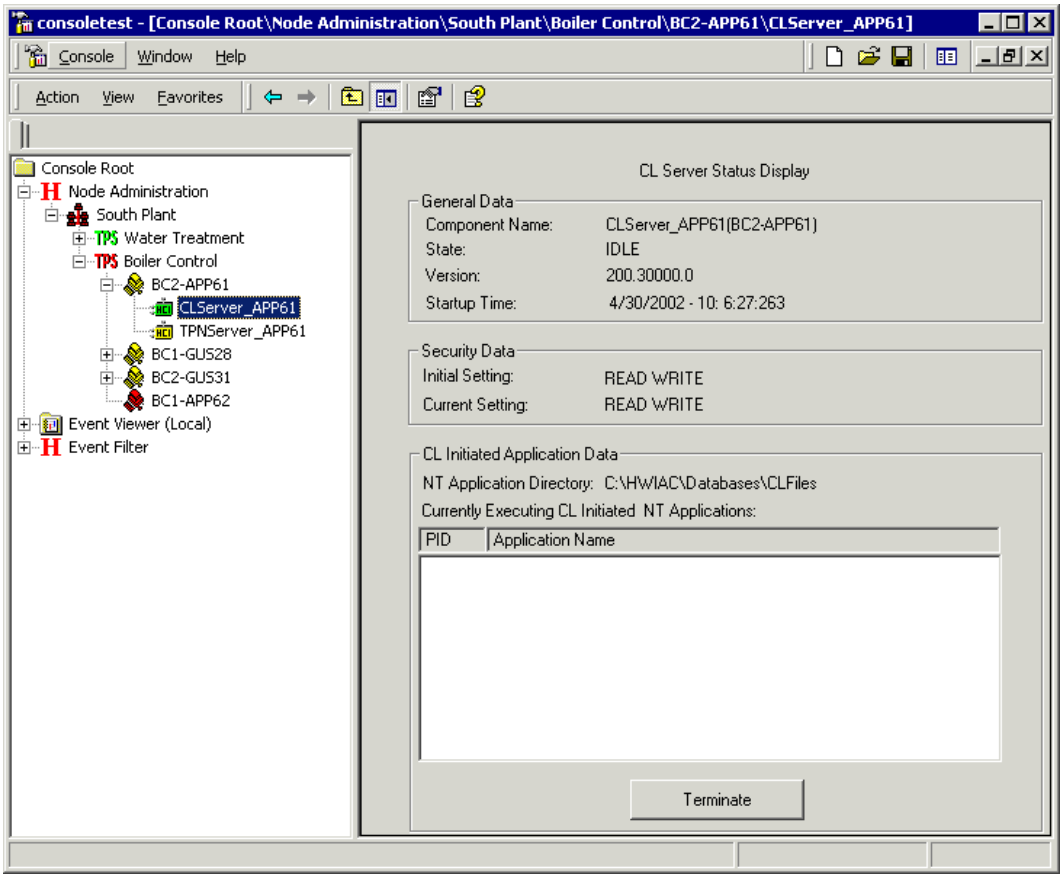
Note: The **Auxiliary Display** *cannot* be invoked in the **Stopped** or **Fail** states.



5.5.1 View the CL Server Status Display

1 Click **Auxiliary Display**.
The **CL Server Status Display** appears as shown. This is also referred to as the **Auxiliary Status Display**.

Note: See Section ' “Auxiliary Status Display - CL Server” on page 41' for a description of the three **Data** sections contained in the status frame.



2 To close the Auxiliary Status Display, right-click the **CL Server** and select **Auxiliary Display** from the menu.

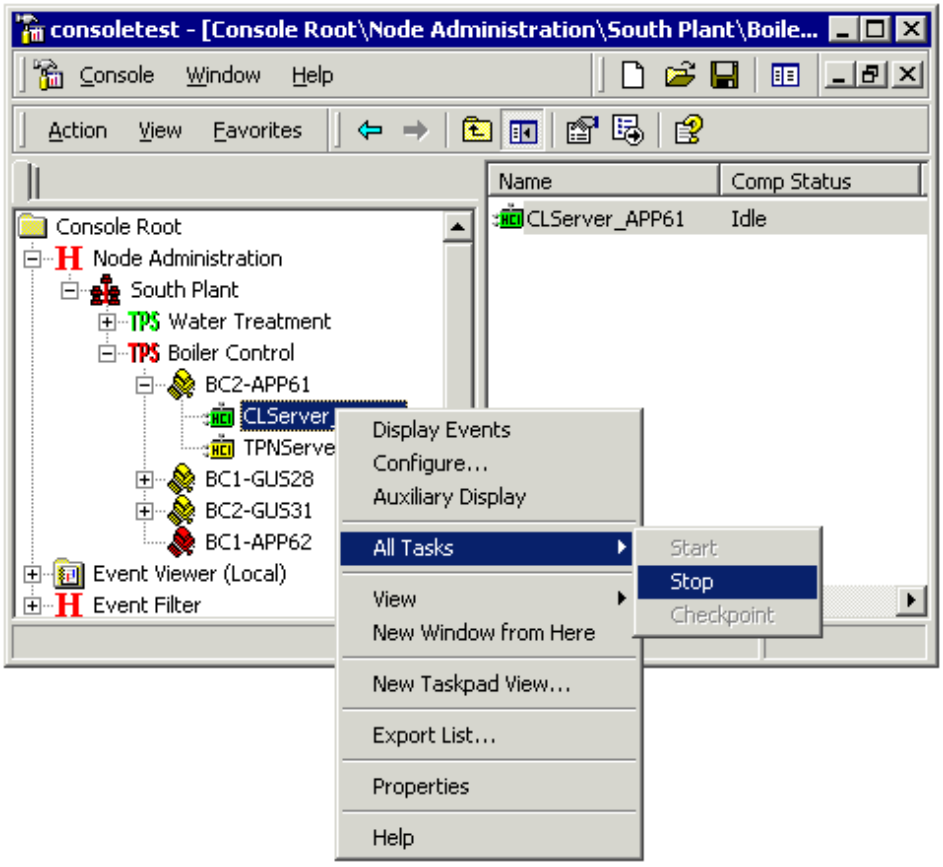
5.6 Shutting Down the CL Server

1 Invoke the System Management Display using one of the methods in the procedure *Starting-up the System Management Display*.

> **Attention**
> You must be logged on as the System administrator or as a user that has shutdown permissions to perform the actions in the following steps

2 Right-click the HCI component that represents the CL Server that you want to shut down and select **All Tasks** > **Stop**

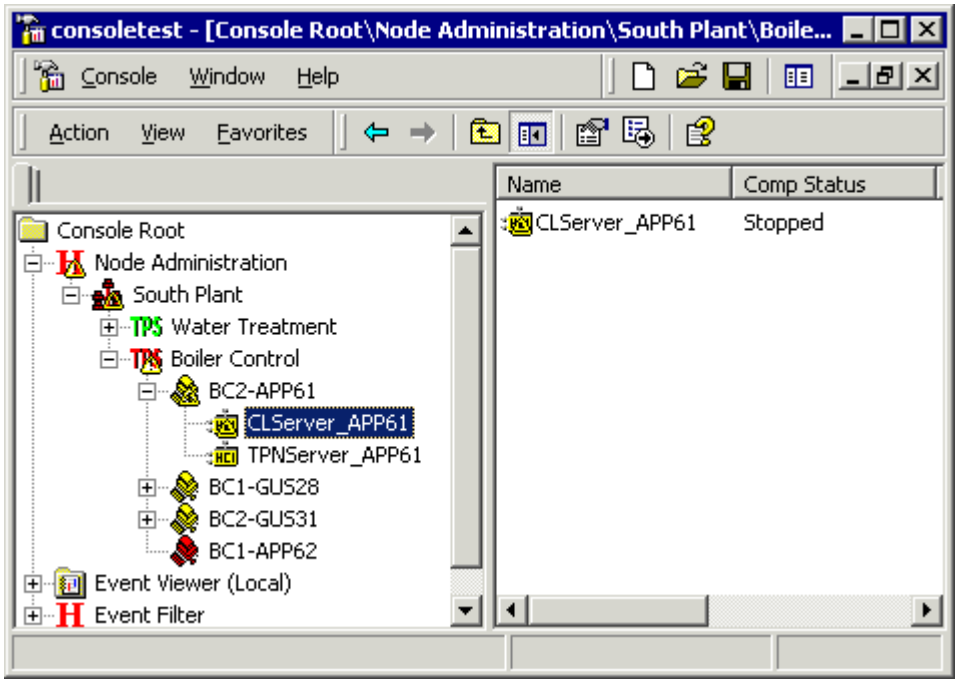
Note: If the CL Server is in the Running or Idle state, it can be shut down from the System Management Display.



5.6.1 Shut down the CL Server

- Observe the CL server transition to a shutdown state.
  The **State** of the HCI component **CLServer_APP61**

  transitions to **Shutting Down** and returns to **Stopped** or **Idle**. The CL Server shutdown is complete.


consoletest - [Console Root\Node Administration\South Plant\Boile...
Console
Window
Help
Action
View
Favorites
Name
Comp Status
CLServer_APP61
Stopped
Console Root
Node Administration
South Plant
Water Treatment
Boiler Control
BC2-APP61
CLServer_APP61
TPNServer_APP61
BC1-GUS28
BC2-GUS31
BC1-APP62
Event Viewer (Local)
Event Filter

5.7 Auxiliary Status Display - CL Server

The CL Server Auxiliary Status Display is divided into three sections: CL Server General Data, Security Data, and CL-initiated Application Data.

Figure 5: CL Server Status Display

5.7.1 General Data section

Component Name	The name of the managed component. The host name is shown in parentheses.

State	The State of the currently active CL Server. The Auxiliary Status display is activated only after the component starts; and therefore, only a subset of the possible HCI states is available for viewing. Due to their transitory nature some of these states will never be shown. The State will be one of the following values: **FAILED**: The server has reached an unrecoverable error and has failed. **IDLE**: The server is operational, and no CL Initiated Applications are running. **INITIALIZING**: The server is initializing. **RUNNING**: The server is operational, and CL Initiated Applications are running. **SHUTDOWNCOMPLETE**: The server shutdown has completed. **SHUTTINGDOWN**: The server is shutting down. **TEST**: The component is performing a self-test. **WARNING**: The server is running, but there is no connection to the TPS Network.
Version	The version number of the currently active CL Server
Startup Time	The time and date the CL Server was started.

5.7.2 Security Data section

Initial Setting	The initial setting of the $XACCESS security.
Current Setting	The current setting of the $XACCESS security.

5.7.3 CL-Initiated Application Data section

Application Directory	The path to the directory where all the CL-Initiated Applications reside. This directory can contain the actual executable (such as, .exe), or a shortcut (such as, .lnk) to an executable in a different directory.
Currently Executing CL-Initiated Applications	The PIDs (Process Identifier), and the names of the currently executing CL-Initiated Applications
Terminate Button	This button becomes active when you select one or more of the CL-Initiated Applications. When this button is clicked, the selected applications will be terminated.

Attention

If the value is displayed as '**?**', then the data are either inaccessible or the CL Server is no longer running.

If the CL Server has failed, the State value will display **FAILED**, and the rest of the values will display '**?**'.

6 Supporting CL Applications

6.1 Application Directory Summary

The information in this subsection is an expansion of the definition of the Application Directory shown on the CL Server Status Display (Auxiliary Status Display). It explains the use of shortcuts, precedence in launching CL-Initiated applications and accessing data using Application Directory.

6.1.1 Using shortcuts

Attention

When a shortcut is created in the Windows Explorer, there is a hidden 'lnk' extension appended to the file name. This extension is not visible in the Windows Explorer. It is visible though, in the old Windows File Manager (winfile.exe). Thus, it is possible to have an executable 'xyz.exe' in the Application Directory, and a shortcut to 'xyz.exe' in the same directory. The real file name of the shortcut would be 'xyz.exe.lnk'. The files will appear in the Windows Explorer as having the same name, but are different files.

6.1.2 Precedence in launching CL Initiated applications

When the CL Server receives a request from the TPN Network, to launch a CL-Initiated application, the following order of precedence is used to determine the application to launch:

1. If the file exists, exactly as typed in the command line received from the CL Block, it is launched. For example, if the file name was 'xyz.exe', and 'xyz.exe' exists in the Application Directory, 'xyz.exe' is launched. If the file name was 'xyz.exe.lnk' (**Note:** The '.lnk' extension will be hidden in the Windows Explorer), and 'xyz.exe.lnk' exists in the Application Directory, the shortcut will be de-referenced, and the de-referenced executable will be launched.
2. If the file does not exist, as typed in the command line received from the CL Block, an '.exe' extension is appended onto the filename. A check is then made to see if the file exists with the '.exe' extension. If it does exist, it is launched. For example, if the file name was 'xyz', an '.exe' is appended onto the file name. The file name 'xyz.exe' is then checked to see if it exists.
3. If the file still does not exist, a '.lnk' extension is appended onto the filename. A check is then made to see if the file exists with the '.lnk' extension. If it does exist, the shortcut is de-referenced, and the de-referenced executable is launched. For example, if the file name was 'xyz', a '.lnk' is appended onto the file name. The file name 'xyz.lnk' is then checked to see if it exists.

6.1.3 Accessing data for CL-Initiated application

Attention

The CL Server and all CL Initiated Applications run under the Experion App user account (a member of the Programs global group). If a CL Initiated Application is accessing a data file, make sure that Experion App has appropriate access to the file and directory where the data file exists. Also, if the Application Directory contains a shortcut to a given CL Initiated Application, make sure that Experion App account (or Programs global group) has, at minimum, read and execute access to the file and directory where the executable exists

6.2 CL Specific Language Support

6.2.1 Supporting Functions

The following background CL calls are available in the AMCL06_2 set in TPS Network release R433 and later:

- **AMCL06$Initiate_Task** — Initiates a Windows application capable of receiving background CL events.
- **AMCL06$Activate_Task** — Sends an activate event to a CL-initiated Windows application.
- **AMCL06$Terminate_Task** — Sends a terminate event to a CL-initiated Windows application, providing a graceful shutdown of the application.
- **AMCL06$Execute_Task_With_Wait** — Initiates an application on the Windows-side running the CL Server.

Initiate_Task

The background CL subroutine **AMCL06$Initiate_Task** is used to initiate a new instance of a Windows application. After the call is made, the background CL suspends execution and enters a wait condition until the Windows application has initiated. The user provides an application name, and a unique application id is returned to the CL block. The CL should store this application id in a CDS parameter or other point.parameter and then complete the execution of the CL block. The application id returned from the subroutine is needed to subsequently activate (**AMCL06$Activate_Task**) and terminate (**AMCL06$Terminate_Task**) the application.

Syntax of the **AMCL06$Initiate_Task** subroutine:

SUBROUTINE AMCL06$Initiate_Task

(Ret_Status : OUT NUMBER; -- Return status of the call

Det_Status : OUT NUMBER; -- Detailed return status

Appl_ID : OUT STRING; -- Application identifier

Cmd_Line : IN STRING; -- Windows application command line

NT_Task_Timeout : IN TIME; -- NT timeout value

Req_Timeout : IN TIME) -- TPN-side timeout value

Refer to Appendix I of the CL/AM Reference Manual for more information about the functionality of the call and for definition of the arguments.

Alternate Approach to Launch Windows-Side Hibernating Applications

The CL Server provides one method for initiating side applications from CL blocks. This is done with the subroutine **AMCL06$Execute_Task_With_Wait**. The approach to launching Windows-side hibernating applications, through the subroutine **AMCL06$Initiate_Task**, is not supported. If this functionality is desired, an alternate approach to implementing hibernation is described in the following section.

The CL Server is capable of running up to ten applications at a given time. These applications execute from the time the CL Server launches them until they terminate normally or exit with an error. Thus, if a given application is launched numerous times, it runs from start to finish each time. By contrast, an application that uses hibernation behaves in a different manner. It is designed to be launched once, and remain persistent in memory between executions. This offers the following advantages:

- The initialization required by the application is performed only once.
- Data remains persistent between executions.
- While the application is hibernating, the CL Server does not manage it. Thus the hibernating application does not count against the maximum of ten running applications supported by the CL Server.
- The shutdown of the application is performed only once.

Alternate Approach

Hibernation can be programmatically simulated by having a CL Block activate a surrogate application on the Windows-side through the CL Server. The surrogate application can be a simple application that has the ability to launch and communicate with hibernating applications.

Surrogate Application

The surrogate application acts as a middleman between the CL Block and the hibernating application. It receives commands from the CL Block in the form of command line arguments. It then uses Windows events to pass the command to the hibernating application. Its purpose is to signal the hibernating application that a command has been requested by a CL Block. It then suspends its processing until it receives acknowledgement that the hibernating application has completed processing the command. It then exits.

The command line, passed from the CL Block, is of the following form:

<surrogate app> <command> [<hibernating app>] <point name>

Where:

Surrogate app – The name of the surrogate application that is launched through the CL Server. This is the application that the CL Block is requesting the CL Server to run.

Command – One of the following commands:

- **Initiate** – Initiate a new instance of a hibernating application. The name of the hibernating application must be passed with this command.
- **Activate** – Send an activate event to a hibernating application. This event will signal the hibernating application to activate its process algorithm.
- **Terminate** – Send a terminate event to a hibernating application to begin its shutdown logic and exit.

Hibernating app (Optional – only needed with the **Initiate** command) - The name of the hibernating application to be launched by the surrogate application.

Point Name– Passed to insure a level of uniqueness for communications between the surrogate application and the hibernating application. The point name is used to create unique events described in the following section.

Unique Events

Windows Events are used as the mechanism for communication and synchronization between the surrogate application and the hibernating application. By this approach, it is very likely that the user will create one surrogate application, and multiple hibernating applications. Thus, unique events must be used to communicate between the one surrogate application and the many hibernating applications. To make the events unique, the point name is used in the event name. Since the point names are unique on the TPN Network, the following convention will create unique event names on Windows:

CL_<Point Name>_[Activate|Hibernate|Terminate|TermCmplt]

Note: the Point Name is passed to the surrogate application from the CL Block.

The following events must be supported:

- **CL_<Point Name>_Activate** – This event is signaled by the surrogate application when it receives the Activate command from the CL Block. This event will signal the hibernating application to activate its process algorithm.
- **CL_<Point Name>_Hibernate** – This event is signaled by the hibernating application when it enters its hibernating state. It is intended that the surrogate application will be waiting on this event to become signaled after it has processed an **Initiate** command or an **Activate** command. By signaling this event, the hibernating application informs the surrogate application that it has completed its task.
- **CL_<Point Name>_Terminate** – This event is signaled by the surrogate application when it receives the **Terminate** command from the CL Block. This event will signal the hibernating application to begin its shutdown logic and exit.
- **CL_<Point Name>_TermCmplt** – This event is signaled by the hibernating application when it has completed its shutdown logic and is exiting. It is intended that the surrogate application will be waiting on this event to become signaled after it has processed the **Terminate** command.

Hibernating Application

The hibernating application is an event driven application that is launched by the surrogate application. This application is launched by a CreateProcess() call done by the surrogate application when it received the **Initiate** command from a CL Block. Upon completing its initialization logic, it enters a hibernating state in which it suspends its process. While in the hibernating state, it waits until one of the following events becomes signaled: **CL_<Point Name>_Activate** event or **CL_<Point Name>_Terminate** event.

When the **CL_<Point Name>_Activate** event becomes signaled, the process algorithm is executed. Upon completing the process algorithm, it signals the **CL_<Point Name>_Hibernate** event to notify the surrogate application that the process algorithm has completed, and it is now hibernating.

When the **CL_<Point Name>_Terminate** event becomes signaled, the shutdown logic is executed. Upon completing the shutdown logic, and prior to exiting, it signals the **CL_<Point Name>_TermCmplt** event to notify the surrogate application that the shutdown has completed, and it is now exiting.

Limitations with the Approach

This approach has the following limitations.

- The subroutine AMCL06$Get_Hiber_Task_Status is not supported. Thus, there is not a provided method for getting the current status of the hibernating application.
- The CL Server and all CL initiated applications run under the ExperionApp account. When the surrogate application does a CreateProcess on the hibernating application, the hibernating application may also run under the ExperionApp account. If the hibernating application runs under the ExperionApp account, a user will not have access to terminate application with tools like the Task Manager. In cases where a hibernating application hangs, the user may have to re-boot Windows to terminate the application.

Attention

The account the hibernating application runs under it at the discretion of the author of the surrogate application. The author has the option of implementing the CreateProcess call to launch a process under a different user.

- The hibernating application is not monitored by the CL Server, and thus will not appear on the CL Server Auxiliary Status Display.

Activate_Task

The background CL subroutine **AMCL06$Activate_Task** is used to send an activate event to a CL-initiated Windows application. After the call is made, the background CL suspends execution and enters a wait condition until the Windows receives and processes the event. The user provides an application id (obtained from **AMCL06$Initiate_Task**) and an event string, which is passed to the application.

Syntax of the **AMCL06$Activate_Task** subroutine:

SUBROUTINE AMCL06$Activate_Task

(Ret_Status : OUT NUMBER; -- Return status of the call

Det_Status : OUT NUMBER; -- Detailed return status

Appl_ID : IN STRING; -- Application identifier

Event_String : IN STRING; -- String passed to application

NT_Task_Timeout : IN TIME; -- NT timeout value

Req_Timeout : IN TIME) -- TPN-side timeout value

Refer to Appendix I of the CL/AM Reference Manual for more information about the functionality of the call and for definition of the arguments.

Terminate_Task

The background CL subroutine **AMCL06$Terminate_Task** is used to send a terminate event to a CL-initiated Windows application. This provides a graceful shutdown of an application. After the call is made, the background CL suspends execution and enters a wait condition until the application has received and processed

the event and terminated its execution. The user provides an application id (obtained from **AMCL06$Initiate_Task**) and an event string, which is passed to the OpenDDA application.

Syntax of the **AMCL06$Terminate_Task** subroutine:

SUBROUTINE AMCL06$Terminate_Task

(Ret_Status : OUT NUMBER; -- Return status of the call

Det_Status : OUT NUMBER; -- Detailed return status

Appl_ID : IN STRING; -- Application identifier

Event_String : IN STRING; -- String passed to application

NT_Task_Timeout : IN TIME; -- NT timeout value

Req_Timeout : IN TIME) -- TPN-side timeout value

Refer to Appendix I of the CL/AM Reference Manual for more information about the functionality of the call and for definition of the arguments.

Execute_Task_With_Wait

The **AMCL06$Execute_Task_With_Wait** subroutine is used to initiate an application on the Windows-side running the CL Server. It can only be called from background CL. After the call, the background CL suspends execution and goes into a wait condition until the Windows-side application terminates normally or terminates because an error was detected.

The following is the syntax of the **AMCL06$Execute_Task_With_Wait** subroutine:

SUBROUTINE AMCL06$Execute_Task_With_Wait

(Ret_Status : OUT NUMBER; -- Return status of the call

Det_Status : OUT NUMBER; -- Detailed return status

Cmd_Line : IN STRING; -- NT-side application command line

NT_Task_Timeout : IN TIME; -- NT-side timeout value

Req_Timeout : IN TIME) -- LCN-side timeout value

Ret_Status and Det_Status

Ret_Status is the return status of the subroutine call. Det_Status (Detail Status) may contain additional information. Values are:

Value	Ret_Status	Det_Status
0	Application terminated normally	Application exit code (defined in the application program)
1	Subroutine argument error	1 = Invalid NT_Task_Timeout 2 = Invalid Req_Timeout 3 = Invalid Cmd_Line
2	CL timeout while waiting for Windows-side application to complete—application aborted	0
3	Unable to communicate with Windows-side	0 = No detail status 4 = Unexp'd connect from Windows-side 5 = Unexp'd disconnect from Windows-side
4	Error getting memory in AM	0
5	LCN/Windows-side connection down	0

Value	Ret_Status	Det_Status
90	Initialization in progress—attempting connection to Windows-side (should only occur during node startup)	0
91	Initialization unsuccessful—unable to acquire internal resources	0
92	Initialization in progress—attempting connection to Windows-side (should only occur during node startup)	0
93	Initialization unsuccessful—unable to acquire internal resources	0
500	Internal error	0
501	Internal produce error	0
502	Internal consume error	0
503	Internal priority error	0
504	Internal message/queue mismatch	0
999	Node not an AMW	0
1001	Application was killed by signal	Signal number
1002	Application name invalid or application not found	NT errno (or 0)
1003	Application timed out (Windows-side timeout)—application aborted	0
1004	Miscellaneous internal error	0
1005	An error occurred while changing priority	NT errno
1006	An error occurred while creating the application's process	NT errno
1007	The application did not have any execute permission set	0
1008	An error occurred while executing the application	NT errno
1009	A miscellaneous NT error occurred in the CDS/CL Server	NT errno
1010	The application command line contained an absolute path, which is not allowed	0
1011	An error occurred while setting up the application's environment	0
1012	Function unavailable	This function of AM/CL is not supported in Experion APP nodes.

Cmd_Line

This argument is passed to the Windows-side where it is interpreted as an Windows command line. It is a string of up to 78 characters containing the application name, not including the pathname, and any application arguments. The application is responsible for interpreting the command line arguments. The application or a link to the application must reside in the directory C:\ProgramData\Honeywell\databases\clfiles.' See the 'Application Directory Summary' section in this document for further information.

Cmd_Line Restrictions

The Windows command line can contain multiple arguments, but the first argument must be the Windows-side application name. The Application name should not include an explicit directory path. Cmd_Line is a CL local string variable, and is therefore limited to a maximum of 78 characters.

Cmd_Line is not processed for Windows shell commands, and therefore does not support items such as:

- Windows shell commands such as .bat files
- Multiple commands separated with ';'
- Applications that have User Interfaces.
- Input/output redirection such as: |, <, >, >>

NT_Task_Timeout

This argument passes a time value to the Windows-side, which is the maximum time allowed for the application to run before it is aborted and an error status returned. A zero time value disables the Windows-side timeout function. Valid time values are in the range from 0 to 24 hours. **Req_Timeout**

This argument contains the maximum time allowed for the CL request to wait for an application completion return before an error status is returned. This timer function is performed on the LCN-side. If a timeout occurs, an Windows-side application abort is requested. A zero value disables the LCN timeout function and is equivalent to an infinite timeout. Valid time values are in the range from 0 to 24 hours.

Req_Timeout Example

```
PACKAGE
--
BLOCK EXECTASK ( GENERIC; AT BACKGRND )
--
%INCLUDE_SET AMCL06
--
LOCAL STRTAPPL : STRING
LOCAL RET_STAT, DET_STAT : NUMBER
LOCAL NT_TMOUT, REQTMOUT : TIME
--
CALL BKG_CHANGE_PRIORITY( LOW )
SET STRTAPPL = 'APP1'
SET NT_TMOUT = 1 MINS
SET REQTMOUT = 2 MINS
--
CALL AMCL06$EXECUTE_TASK_WITH_WAIT (RET_STAT,
& DET_STAT,
& STRTAPPL,
& NT_TMOUT,
& REQTMOUT)
--
SEND: 'EXIT CODE = ',DET_STAT
--
IF (RET_STAT <> 0.0) THEN
& SEND: 'ERROR: STATUS = ', RET_STAT
--
END EXECTASK
- -
END PACKAGE
```

7 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.

7.1 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.

7.2 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.

7.3 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.

7.4 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.