

# **Control Architect**

User guide 2024

# **Table of Contents**

Introduction	6
Welcome	7
What's new	8
Product Roadmap	20
Getting Started	21
System requirements	22
Getting help	23
Control Architect licensing	24
How to purchase Control Architect	25
Common Tips & Tricks	26
Application Themes	30
List View Filtering	31
Common Control Architect Tasks	33
Overview of the user interface	34
Control Architect window	35
Ribbon tabs	39
Tools menu	40
History menu	43
View menu	44
File Menu-Options	46
Ribbon menu bar	49
Mongo Server Options	51
Solution View	58
Properties view	59
Code view	60
Evaluation view	61
PKS Configuration Views	62
PKS Views Introduction	64
Open existing PKS Configuration project	65
Migrating to new document Server	67
PKS Configuration Import Options window	68
Getting Started Using PKS Views	
Opening a new PKS Configuration Project	74
Managing Configuration Projects MRU List	75
Batch Printing	78
ChartView	81
Annotation View	84
Working with Ink	
Ink Annotation options	
Ribbon menu bar functions	
ChartView Overview	
ChartView Help Tips	
XPM Logic View Overview	
XPM Logic Tool windows	
XPM Logic View	
ChartView Ribbon menu bar	
NonCEE References	
Document References	
Working with Chart Views	
HMIWeb View	
Using the HMIWeb View	
HMIWeb Graphic Capture	
HMIWeb Graphic Capture Overview	
DataView	
DataView Overview	
DataView Filtering	
DataView Advanced Find Overview	
DataView Grouping	126

Container Grouping	127
Data Export Wizard	128
Data Export Welcome Page	129
Data Export Media Type Page	
Data Export Parameter Scope Page	
CM Parameter Scope Page	
All Parameter Scope Page	
Data Export Parameter Formatting Page	
Data Export Export Results	
PeerMap	
PeerMap Overview	
Peer Map Copy/Paste to Excel	
DataView Ribbon menu bar	
SCM# View	
Working in the SCM# View	
SCM# Language View Overview	
SCM# Language Source Overview	
SCM# Sequence Language Overview	
XmlView	
XmlView Overview	
Configuration Asset Model Generation	
Parsing Failure	
PKS Control Graph	
Using the PKS Control Graph	
Control Graph View	166
Control Graph Vertex Block	169
PKS Control Graph View Examples	171
Control Graph Options	175
Importing Honeywell EB, FSC, SCADA, and Triconex Point Data Sources	180
Working with LCN & SCADA Views	182
Import Honeywell LCN EB and SCADA Point Files	184
Using Picture Editor MPRINT to create DX Files	188
Importing Honeywell FSC(SM) Export Files	190
Getting Started using FSC(SM) Integration	
FSC PLC Logic View	
toolbar	
Using TPS Builder SqlServer Synchronization	
Example Configuration Views	
Importing Triconex PLC Export Files	
Getting Started using Triconex Integration	
TRICONEX PLC Logic View	
toolbar	
Browse for Server Database:Collection window	
Stats View	
Stats View Ribbon functions	
Bar Chart Overview	
Summary View	
Block Usage View	
XPM Slot Usage View	
Example Copy/Paste to Excel	
Block Types Editor View	
Class Types Editor View	
LCN Hardware View	
Visual Structure	
CM Duplicate View	
Duplicate View window	
MongoDB Document Server	258
MongoDB Server Service	259
Connecting to Remote Server	260
Server Maintenance Overview	266

Server Maintenance View	267
Managing Server Database	269
Moving Server Database	270
Maintaining Multiple Databases	272
Working with Mongo Shell	274
Mongo Database Packaging window	277
MongoDB Database Unpackaging window	279
Project Data Map	281
Project Data Map Options	284
ICG PeerView	285
Configure the View	287
Export Server Data	288
Configure the peer nodes	290
Exporting to Excel	292
Workflow Manager	
PolicyCop	
Getting Started with Policy Validation	
PolicyCop Workflow View	
Comparison Results View	
Using Rule Exclusion	
PolicyCop Rules	
PolicyCop Rule Authoring	
Rule Authoring View	
Ribbon Tools tab	
Ribbon File tab	
Ribbon Edit tab	
Ribbon Help tab	
Auto-Rule Generation	
Solution view	
Using the Solution view	
Solution Packaging window	
Solution Unpackaging window	
Installed project item Templates	
Bulk Build/Edit	
Workflow Task View	
Create Control Conversion Project Workflow	
Specify Solution Project options	
Specify Excel DataSource File options	
Specify Target Properties	
Specify Generation options	
Generate Solution Project	
Control Conversion views	
Getting Started Using Control Conversion Views	
Control Property view	
Control Property Overview	
Importing Parameter References	
Import Wizard DataSource selection	
Import Wizard choose XML parameters	
Import Wizard choose EB parameters	
Import Wizard specify parameters Template	
Import Wizard completion	
Control Property Template results	
Example Excel Workbook Data Source	
Control Binding view	
Working with Control Property Bindings	
Control Binding expressions	
Control Binding Expression examples	
Custom API Methods	
NonCee References	
Control Binding Evaluation	

Control Architect Help Documentation	
Constants Editor	377
Custom Variables	378
Code Snippets	379
Binding to Block Pin connections	381
Block Building	382
Disabling Bindings	383
Control Conversion view	385
Control Conversion View	386
Defining Control Conversion bindings	392
Excel Data Source filter expressions	394
Data Collections	397
Parsing EB files to Excel	400
Data Source Control Graph View	401
Control Graph Overview	402
Control Graph References	404
Embedding Modules	405
History	407
History View	408
Ribbon menu bar	410
History Configuration window	412
Compare View Overview	414
Compare View	415
Report View ToolBar	418
Compare Entity List	420
Compare Excluded Parameter List	422



# Welcome

Welcome to AST's engineering tool for Honeywell Process Control Systems including LCN, SCADA, and Experion Process Knowledge System (EPKS). Control Architect provides a suite of engineering tools and functions that reduce overall engineering costs during project execution and provide powerful analysis and documentation tools for life cycle management of PKS based control systems.

This help documentation is designed so you can quickly learn Control Architect as a new user or enhance your knowledge as a regular user.

#### What's new

This release of Control Architect provides powerful tools to aid in control strategy analysis and validation along with tools for quickly building control strategies using existing Control Modules as a typical or Template.

#### Features include:

- PKS Control Builder exported (Cxxx,ACE) configuration importing to a Configuration Asset Model to facilitate control strategy analysis and documentation
- PKS Control Edge is supported
- SCADA, Triconex, and FSC configuration import
- LCN EB (Exception Build) configuration import. ENIM/EHPM are supported.
- HMIWeb and LCN Schematic DX file import
- · Exporting of configuration data to Excel
- Control Module chart viewing similar to the Control Builder chart view with enhanced features and functions
- SCM function block conversion to SCM# Language script for straightforward analysis of complex SCM's
- Control Module Chart View Ink surface annotation to facilitate white board collaboration amongst users
- Full external references for all function blocks and parameters including references to CAB and SCM alias references
- Visual peer analysis within community server nodes including ACE, C300, and FIM
- Visual InterCluster parameter analysis of control strategies between EPKS server nodes and communities
- Policy validation engine (PolicyCop) that provides powerful rule authoring and execution on Cxxx/ACE project XML configuration files
- Automatic policy rule generation based on existing Cxxx/ACE Control Modules to be used as a Template or typical
- Workflow activities used for Control Conversion tasks to convert Honeywell Exception Build (EB) files to Excel
- Control Conversion views that support property (parameter) generation and binding to external Excel Data Source types to support Control conversion tasks such as migrating Hiway based configuration to C300 or building/editing Control Modules based on data residing in an Excel Workbook file
- Workflow activity that generates new Control Module configuration based on data binding to external Excel Data Sources
- Workflow activity that provides batch editing of Control Module configuration
- Integrating reporting with export to media types such as Microsoft Word and Adobe PDF
- Project configuration comparison for changes



It is important that all users report any and all defects so that we can continue to provide a high quality robust application for engineers. **Thanks to users and organizations** that have contributed both in reporting defects and proposing new enhancements that result in reduced engineering effort and life cycle costs for integrated Honeywell Control systems.

# What's New in Control Architect Release 24.30.x - {September 8, 2024}

# **New LCN Network Hardware View**

Added a new LCN Network Hardware layer view to the configuration views. New hardware views are being developed and deployed for all supported Control System platforms. These same hardware views will support the upcoming release that will include the ability to reserve I/O for future project usage.

## **Support for PKS Control Builder EHPM**

LCN EHPM configuration exported through Control Builder is now supported.

### What's New in Control Architect Release 24.20.0 - {July 29, 2024}

#### PKS CDA and SCADA History is now supported

When importing PKS Control Modules (CDA) and SCADA point data the history configuration is also now imported and viewed in the configuration Data View along with the PI and PHD History configuration. A new report has also been implemented to generate history parameter data to an Excel Workbook.

## What's New in Control Architect Release 24.10.1 - {June 10, 2024}

#### MongoDB Server is upgraded to 6.0.10

Installer now upgrades the existing MongoDB Server 5.0.12 installations to MongoDB Server 6.0.10. Existing 5.0.12 Mongo Server installation is also uninstalled during installation of Control Architect.

#### **Support for DMC configuration asset**

DMC configuration is now supported as an asset when importing configuration assets. DMC references are now shown in Chart views.

### **Enhanced Control Conversion Functions**

Scada Control Conversion binding now supports binding to the parameter names "STATETXTn" where n = 0-7 to form the required Scada Status 'STATEDES' parameter value.

#### Mongo Database Packaging now supported for Remote Servers

MongoDB Database packaging is now supported for all remote server applications including the Mongo Atlas Cloud server.

#### **Export Wizard now supports integration to Honeywell TPS Builder application**

Export Wizard now supports integration with Honeywell TPS Builder to support exporting of specific container structure and parameter data. This function is intended for those customers that use TPS Builder for maintaining control systems developed using hierarchical folders and containing modules.

#### **PKS Control Module Duplication enhancements**

PKS Control Module duplication function now supports changing BlockName values such may be the case where I/O Channel blocks are renamed to match the Instrument device.

#### **Other Enhancements**

Numerous other small enhancements including adding Channel and Controller RTU parameter data to the Chart View Block surface parameters when importing SCADA data into a configuration snapshot.

#### **Defect Removal**

Many defects reported by others and found during testing and Control System project usage have been removed. View the detail release notes for specifics on defect fixes as well as new functions and enhancements.

## What's New in Control Architect Release 23.30.0 - {August 28, 2023}

#### **Enhanced Control Conversion Functions**

Added support to Control Conversion functions for Parameter names that start with a '\$' character as typically found in LCN Parameter data. This allows us to reference this particular Parameter or Property type in a script.

A Bulk Edit generation function has been added to the Data Collection view generated when executing a Control Conversion task. This menu option generates a Control Builder Excel compliant data set that can be pasted from the Windows Clipboard into an Excel (\*.XLS) file. The user simply has to rename the default "Sheet1" Worksheet name to "Parameters".

Enhanced Control Conversion importing of HW Template or Based On configuration entity parameters such that Parameters are properly typed as a 'Double' vs an 'Integer'. Often parameters such as trip points and other double types may be incorrectly set as integer types when they should have been double types. The user has to catch this type of error when the Data Collection parameter values are inexplicably being rounded, which is the case when a double value is converted to an integer. There is now a list of often encountered Double type parameters that is now used to qualify whether a number value is a true Double type.

# **Defect Removal**

Many defects reported by others and found during testing and Control System project usage have been removed. View the detail release notes for specifics on defect fixes as well as new functions and enhancements.

# What's New in Control Architect Release 23.20.0 - {March 28, 2023}

#### **Enhanced Control Conversion Functions**

Enhanced the Control Conversion Data Collection View to always show the 'New' entity items to be created regardless of whether any Parameter data changes other than 'Name' were detected when copying the data to the Windows Clipboard.

## **Defect Removal**

Many defects reported by others and found during testing and Control System project usage have been removed. View the detail release notes for specifics on defect fixes as well as new functions and enhancements.

## What's New in Control Architect Release 23.11.0 - {March 10, 2023}

#### **Enhanced TPS Builder synchronization during Import**

During import 'Custom' and 'Identification' Parameters are now imported to the generated configuration Control Module entity.

#### **Enhanced Control Conversion Functions**

Enhanced the Control Conversion for SCADA type generation to support the "BLOCK\_DELETE" command being used in a 'NAME' Control Binding expression. If the expression evaluates to "BLOCK\_DELETE" then the Entity will not be generated.

A new function named "ExistsInColumn" was added. This function is used to execute a query on the specified Worksheet in the Excel Workbook that contains the data being processed. It will lookup and try to find the specified value in the specified column. This function will return 'True' if the specified query statement results in a data item being found. This function can be referenced when developing a Control Binding expression.

## **Enhanced Data View Data Export function**

Enhanced the Data View data export function.

• UI navigation and content has been changed to reduce complexity in choosing the parameter scopes to export.

- The default Parameter scope is now defaulted to "Embedded Params", which provides access to all parameters available for a particular Entity or Control Module.
- Parameter list file paths are now saved to a Most Recently Used (MRU) list that is saved between sessions. When clicking the dropdown menu button any previous used Parameter list files will be available as a menu item to choose vs having to browse for the file path.
- There is now an option to exclude the containing Control Module params for SCADA & LCN types. Using this option will now export a single row of data for these template types.

#### New Configuration Data View Report

Generates a report that includes all external references to an Entity. This report data is copied to the Windows Clipboard for pasting into an Excel Worksheet. External references that are reported include I/O Modules, Graphics, Application Source code { LCN CL, CDA CAB }, other Control Strategy Parameter references, etc.

#### **Other Enhancements**

View the detail release notes for specifics on other new functions and enhancements.

#### **Defect Removal**

Many defects reported by others and found during testing and Control System project usage have been removed. View the detail release notes for specifics on defect fixes as well as new functions and enhancements.

## What's New in Control Architect Release 22.41.0 - {December 8, 2022}

#### **Enhanced Comparison Report Export**

Enhanced Comparison Report exporting to Excel function such that Excel sheets [NewEntities] and [DeletedEntities] are now created.

- The sheet data also now includes the Entity description.
- When viewing a previously saved Comparison report you did not have the option to export this to Excel. Exporting to Excel is now supported.

#### MongoDB Server upgrade to version 5.0.12

The installed Mongo Server is upgraded to a more recent version during installation.

### **Enhanced Data View Data Export function**

Enhanced the Data View data export function.

- UI navigation and content has been changed to reduce complexity in choosing the parameter scopes to export.
- The default Parameter scope is now defaulted to "Embedded Params", which provides access to all parameters available for a particular Entity or Control Module.
- Parameter list file paths are now saved to a Most Recently Used (MRU) list that is saved between sessions. When clicking the dropdown menu button any previous used Parameter list files will be available as a menu item to choose vs having to browse for the file path.
- There is now an option to exclude the containing Control Module params for SCADA & LCN types. Using this option will now export a single row of data for these template types.

#### **Other Enhancements**

View the detail release notes for specifics on other new functions and enhancements.

#### **Defect Removal**

Many defects reported by others and found during testing and Control System project usage have been removed. View the detail release notes for specifics on defect fixes as well as new functions and enhancements.

## What's New in Control Architect Release 22.30.1 - {August 31, 2022}

#### **New Control Edge Support**

Control Architect has now been validated for UOC Control Edge. Configuration Import and Views are fully supported for UOC controllers configured as a CEE Peer node.

#### MongoDB Server upgrade to version 4.4.1

The installed Mongo Server is upgraded to a more recent version during installation.

#### New Honeywell FSC (Safety Manager) Integration using Modbus PCDI/SCADA

Control Architect now fully supports integrating Safety Manager Modus configuration that utilizes Scada Modbus and PCDI. When importing Safety Manger exported configuration the references between Modbus and PCDI Alias addresses are fully resolved and displayed as connection references when viewing referenced strategies in their Chart views. References that cannot be resolved are indicated as errors during import, which in itself provides a validation to the addresses used in both your Scada and PCDI Channel blocks.

#### **New Interlock Function Block View**

A new Interlock Function Block view has been added to the Property View of a Function Block. This new view visualizes the internal Gate configuration.

## **Other Enhancements**

View the detail release notes for specifics on other new functions and enhancements.

#### **Defect Removal**

Many defects reported by others and found during testing and Control System project usage have been removed. View the detail release notes for specifics on defect fixes as well as new functions and enhancements.

## What's New in Control Architect Release 21.30.1 - {October 1, 2021}

#### **Triconex PLC Viewer Enhancement**

Added new PLC Program Logic viewer that is accessible from either the project configuration Data View or a Program Logic ChartView block. Triconex PLC assets that have been imported into a project Collection snapshot can quickly be viewed from a new drop-down button located in the Ribbon Tab for the Data View. This new view that replaces the original single Program Logic view allows the user to view all programs and their variable usage associated with a PLC Controller asset from within the same view.

#### **Mongo Compass Installation**

Mongo Compass is now installed during installation to provide users another application for administering their MongoDB installations.

#### **Other Enhancements**

View the detail release notes for specifics on other new functions and enhancements.

### **Defect Removal**

Many defects reported by others and found during testing and Control System project usage have been removed. View the detail release notes for specifics on defect fixes as well as new functions and enhancements.

## What's New in Control Architect Release 21.21.1 - {September 15, 2021}

## **Triconex Integration Support**

Added new PLC Asset view that is accessible from the project configuration Data View. Triconex PLC assets that have been imported into a project Collection snapshot can quickly be viewed from a new drop-down button located in the Ribbon Tab for the Data View. Unlike viewing a single PLC Program logic and its variable set from within a Chart View block, this view allows the user to view all program logic and their variable usage associated with a PLC Controller asset.

#### **Defect Removal**

Many defects reported by others and found during testing and Control System project usage have been removed. View the detail release notes for specifics on defect fixes as well as new functions and enhancements.

## What's New in Control Architect Release 21.20.1 - {April 17, 2021}

## **Triconex Integration Support**

This release includes new **Triconex integration** support.

- Users can export TriStation project configuration for importing into a Control Architect supported Mongo Database:Collection configuration snapshot.
- Users can now visualize in their Chart Views Modbus connectivity between PCDI Channel blocks and SCADA Point blocks and a Triconex PLC asset.
- Users can view their PLC Program Logic in a new Logic View with variable usage/navigation and property views.
- Importing of Triconex project assets provides full reconciliation of all Modbus connectivity, which provides an important validation process for the DCS configuration.

#### **Importing**

The configuration Import function used to import other assets such as SCADA, EB, etc. has been enhanced to support importing Triconex assets that were exported from the TriStation application.

#### **Defect Removal**

Many defects reported by others and found during testing and Control System project usage have been removed. View the detail release notes for specifics on defect fixes as well as new functions and enhancements.

# What's New in Control Architect Release 21.11.1 - {March 1, 2021}

### **Application Theme Support**

This release includes new theming support. The default theme "Dark" is installed by default. Users can choose the option to synchronize System Theme settings with the application theme such that if the user changes his system theme from Dark to Light, then the application will automatically select a "Light" theme.

## **Discontinued Support for Windows 7**

This release discontinues support for the Windows 7 Operating System

## MongoDB Server Upgraded to 4.2

This release installs a more recent version of MongoDB Server. Control Architect now supports MongoDB Version 4.2.

A new Log View function has been added to the MongoDB Server Options page that allows a user to view the cached log items for a local or remote MongoDB Server. This shall be used when troubleshooting MongoDB Server connection issues.

#### **Usability Improvements / Refactoring**

#### Application Wide Refactoring

Control Architect was refactored extensively to not only support the ability to set a theme, but to also provide consistency throughout the various views.

#### **Chart View**

Enhancements to support ChartView connections.

#### Control Graph View

Improved block adorner images to provide a more modern technical look and feel that scales well with the Control Graph zoom level.

#### **Configuration Views**

Data View grouping functions have been enhanced significantly to improve both performance and data grouping.

- All group types are now displayed in a hierarchical tree format.
- The grouped views now support the same "Filter" functions as the flat List view.

#### HMIWeb View:

- The function to show the static content of the selected process graphic has been changed to use Win10 installed Edge Chrome based browser as the content viewer.
- The HMIWeb View has been updated to display the File Version for the graphics.
- The copy function Ctrl+Shift+C key has been added, which copies only the summary data for the selected process graphics.
- Graphics when exported now retain their original imported File date.

#### DSA:

DSA support has been added to the configuration Data View that allows you to navigate to other referenced configuration projects. Chart Views that reference an Entity in another configuration project (PKS Server) that is being accessed through DSA will now have its configuration project opened in a new set of configuration views focused on its containing Chart View.

## **MongoDB**

The Mongo Server status LED located in the status bar of the main Application window has been updated to now reflect the status of a remote connected Mongo Server whether that is another Server in your local Network or a Mongo Atlas cloud connection.

#### **Importing**

The PKS configuration Import function has been enhanced to support automatically retrieving assets from other existing Collections. This helps to reduce efforts in creating a complete configuration model where certain types of assets such as the PKS Asset model and others may not change in between generated configuration Collection snapshots. These new import options also allow users to copy existing lnk Annotations and Control Graph View states to the new configuration Collection snapshot.

## **Defect Removal**

Many defects reported by others and found during testing and Control System project usage have been removed. View the detail release notes for specifics on defect fixes as well as new functions and enhancements.

# What's New in Control Architect Release 20.10.1 - {January 20, 2020}

## **Mongo Database Optimization**

Improved performance of Mongo Server connectivity by implementing a cached Database Connection Pool. This Connection Pool will eliminate the need to authenticate anytime a new database operation is requested will should improve performance especially over network connections.

#### **Usability Improvements**

# PKS R520 Support

Control Architect now supports PKS R520 release when generating PKS compliant configuration files. The new release enumeration R520Plus will provide support up to Honeywell planned release of R600 unless a schema change occurs prior to R600.

# HMIWeb View Browser View

Enhanced HMIWeb View Browser View to support a classic Window style vs the Popup view. A Popup view is always attached to the window where it originates from while a classic Window is movable and detachable from the originator.

#### Peer Map View

Enhanced Peer Map View to support a classic Window style vs the Popup view. A Popup view is always attached to the window where it originates from while a classic Window is movable and detachable from the originator.

#### Server Maintenance View

Added new data column to MongoDB Server Maintenance View that shows the timestamp of the configuration files contained in the Collection. The configuration file time stamp is stored as MetaData for a Mongo Database, which is added during import.

#### **Defect Removal**

Many defects reported by others and found during testing and Control System project usage have been removed. View the detail release notes for specifics on defect fixes.

## What's New in Control Architect Release 19.30.3 - {November 7, 2019}

#### **Mongo Database Optimization**

Queries executed in the Mongo Database collections have been optimized to use more effective text collation searches.

#### **Usability Improvements**

#### Chart View Graphic Reference Navigation

Users can now navigate a graphic reference in a Chart View block. Mouse double-clicking a graphic reference item will automatically navigate to the HMIWeb view tab in the configuration views and show the Browser view for the graphic. The graphic Browser view is no longer hidden when switching away from the HMIWeb tab allowing the user to quickly see the graphic content when double- clicking a graphic reference item in a Chart View.

## Chart View Reference Hyperlink Navigation

Chart View external reference tags when toggled to show its reference list now allows navigation of the list items using a single left mouse-click vs having to double-click to navigate. External Reference View list items also now support Hyperlink navigation.

#### **Defect Removal**

Many defects reported by others and found during testing and Control System project usage have been removed. View the detail release notes for specifics on defect fixes.

#### **Enhanced PolicyCop Rule sets**

Enhanced PolicyCop rule sets are available for validating PKS configuration.

## What's New in Control Architect Release 19.30.1 - {October 7, 2019}

## **Enhanced Batch Print**

Batch printing last used options are now saved and restored.

# **Enhanced HMIWeb Capture Tool**

Tool now supports combining generated PDF graphic snapshots into a single PDF Book file.

# **Enhanced SCM/RCM Chart View**

- Chart view now shows the Handler description when hovering the mouse over the Handler Tab header.
- Chart view rendering has been improved for R430+ visuals.
- CDB blocks properties now includes its custom parameters.

#### **Enhanced Control Conversion**

- Conversion now uses the target Property type, which is controlled by the user versus having to infer the type based on the Source property.
- Values such as "----" are now converted to a Double.NaN type as would be typically found in a Honeywell LCN data set.
- Empty values are now converted to a Double.NaN type.

#### **Enhanced Document Search Function**

The document search functions have been enhanced to show an integrated "Find" view that by default is visible when the document is first opened. It can also be opened using the Ctrl+F keyboard command if the search view is closed.

## **Defect Removal**

Many defects reported by others and found during testing and Control System project usage have been removed. View the detail release notes for specifics on defect fixes.

### What's New in Control Architect Release 19.21.1 - {June 24, 2019}

## **New Batch Print**

Batch printing is now supported for configuration documents including, Chart View content, CAB, SCM#, CL source views and LCN Logic/ Device Control Gate views.

#### **R510 Qualified**

Honeywell R510 configuration schema is fully supported.

## **Defect Removal**

Many defects reported by others and found during testing and Control System project usage have been removed. View the detail release notes for specifics on defect fixes.

# What's New in Control Architect Release 19.20.0 - {April 22, 2019}

#### **Enhanced Compare Function Export**

Enhanced Compare function to support exporting results data to Excel without using the generated Report View export function. The exported data is generated into the specified Excel Workbook file that will create the Worksheet tabs { Parameters, Attributes, Connections }.

#### Support for Mongo Atlas

Added support for user to specify address of remote MongoDB Server to connect with.

- User can specify to connect with a local Intranet Mongo Server
- · User can specify to connect with an Enterprise Atlas Cloud Cluster

#### **Defect Removal**

Many defects reported by others and found during testing and Control System project usage have been removed. View the detail release notes for specifics on defect fixes.

#### What's New in Control Architect Release 19.10.3 - {February 25, 2019}

#### Version numbering changed

Control Architect version numbering has changed to better reflect the calendar year that the version is released. Starting in January 2019 all releases issued during the calendar year will have their version number prefixed with "19" while the second part of the version number continues to reflect the quarter of the year where 10-19 reflects quarter 1 and 20-29 reflects quarter 2, etc.

#### **Defect Removal**

Many defects reported by others and found during testing and Control System project usage have been removed. View the detail release notes for specifics on defect fixes.

#### **Usability Improvements**

Below are highlights of improvements and new functions added to this release. Many usability improvements and new functions have been added, which can be viewed in the detail release notes that covers the preview releases starting with 5.23.2.

- Control Conversion supports a new method LookUp(...) that allows a user to reference a Worksheet in the current Workbook file being processed or an external Workbook file to lookup a value for use in developing a Control Binding target value. Prior to this new Method users had to copy or move data into the Worksheet where the primary data was located so that the Control Binding execution had access to the data for developing a target value during conversion.
- Chart View support has been updated for R510+ PKS release
- Control Binding Template views now supports reporting for Control Binding details. A new Report button when clicked will generate a report of all Control Binding details that can be printed or saved to other media type including PDF.
- PHD and PI history data is now supported during importing of configuration data. PHD and PI History details are displayed in the DataView as an additional data column.

#### What's New in Control Architect Release 5.23.1

#### **Defect Removal**

Many defects reported by others and found during testing and Control System project usage have been removed. View the detail release notes for specifics on defect fixes.

## **Usability Improvements**

Many changes to usability have been implemented to improve the user experience including enhancements to PolicyCop, Control Conversion tasks, Chart View rendering, and Solution views.

- PolicyCop now saves the last enabled set of Rules for restoring when PolicyCop is launched
- Solution View now supports Text TXT type files
- Chart View IOChannel Reference blocks now show their virtual Channel Block parameter values on the Block surface
- Chart View SCM rendering has been updated for R430+ PKS releases to align with the Control Builder visuals
- Control Views now support Copy/Paste between Template file views.

## What's New in Control Architect Release 5.22.1

### **Defect Removal**

Many defects reported by others and found during testing and Control System project usage have been removed. View the detail release notes for specifics on defect fixes.

## MongoDB Database Packaging/Unpacking

Users can now package up the logical databases contained in a physical MongoDB database that is connected to the Mongo Server for deploying to other users or simply as a backup.

## **Report View Enhancements**

The Report View has been redesigned with a newer navigation interface and a new Search function that allows users to search for content within the generated reports.

#### **Solution View Enhancements**

The Solution View now supports multiple opened Solution workspace files.

#### **LCN Schematic Import Enhancements**

The LCN Schematic DX File import has been enhanced to provide a consistent display of LCN Schematic references when viewed in the HMIWeb configuration view for the project. Entity.Parameter references are now related to the Subpicture where used as indicated by the "ShapeSource" data column.

Schematic Variable usage now displays the proper LCN Collection Rate. Generated reports will also show Parameter update rates and usage.

## **Control Conversion Enhancements**

Enhancements have been made to provide visual cues when BaseOnFilename file paths do not match between the Control Property and Control Conversion templates.

Default sorting has been added to the Control Property and Control Binding views.

#### **Configuration View Enhancements**

A new refresh button has been added that allows a user to quickly refresh the configuration Asset model without having to close and reopen the configuration project as may be the case when importing additional configuration assets.

#### What's New in Control Architect Release 5.21.0

#### **Defect Removal**

Many defects reported by others and found during testing and Control System project usage have been removed. View the detail release notes for specifics on defect fixes.

## **Workspace Solution Packaging/Unpacking**

Users can now package up their Workspace solutions to a \*.package file that can be used to deploy to another system or simply be used as a backup.

## **Control Graph Enhancements**

Control Graph options view has been redesigned to categorize often used options vs. less used (advanced) options.

#### **Control Conversion Enhancements**

Control Conversion usage has been redesigned to provide a more intuitive flow of data and commands. Saving a generated Data Collection now automatically invokes the Workflow Manager Process view ready for task execution.

### **Peer Map and Configuration Stats View Enhancements**

Peer Map and Stats View data are now saved to the project configuration database. This saved data is automatically loaded when the view is requested instead of generating the data for each view request.

## What's New in Control Architect Release 5.10.4

## **Defect Removal**

Many defects reported by others and found during testing and Control System project usage have been removed. View the detail release notes for specifics on defect fixes.

#### **Mongo Server Management**

Mongo Server management functions have been added to support:

- Maintaining an MRU list of attached MongoDB Databases
- Quickly attaching to a MongoDB Database from the MRU list
- Browsing for a folder path that contains a physical MongoDB Database for attaching to the Mongo Server
- Creating a new MongoDB Database for attaching

#### **Installer Improvements**

The installer now preserves the last attached MongoDB Database when upgrading.

## **Control Conversion Enhancements**

• Warnings are now generated if an embedded BlockName change is not valid in the required CMname.BlockName format.

View the Release notes for details on enhancements that have been made to support Control Conversion tasks.

## **Control Graph Enhancements**

• Vertex node (Blocks) expansion now show connections to existing nodes in the graph without having to expand.

- Control Graph optionally supports showing relationships to Graphics.
- The Control Graph view can now quickly be copied as an image to the Windows Clipboard for pasting.

#### **Other Improvements**

- Configuration import of PKS Scada now supports the Accumulator point type.
- SCM and RCM SCM# conversion now supports writing of PHASE block parameters for those PHASE blocks that are not assigned to a Handler for execution.

#### What's New in Control Architect Release 5.10.0

#### **Defect Removal**

Many defects reported by others and found during testing and Control System project usage have been removed. View the detail release notes for specifics on defect fixes.

#### **PKS Alarm Groups**

Control Architect now supports Alarm Groups that can be selected as an additional Model asset when importing PKS Server configuration.

## **PKS Asset Model**

Control Architect now supports the PKS Asset Model that can be selected as an additional Model asset when importing PKS Server configuration.

#### **Advanced Query for configuration data view**

An advanced query function has been added that allows you to combine multiple filter items using boolean And/Or functions for filtering the configuration Data view.

#### **Control Graph View enhancements**

- Control Graph optionally supports references to graphics. A Graphic is displayed as a block in the generated graph with its connected Entity references.
- A new button has been added to the Toolbar in the view to allow quickly copying the Control Graph to the Windows clipboard for pasting.
- References to existing blocks in the view from blocks that have not yet been expanded are now shown.

#### Solution View enhancements

The Solution View now indicates when a file system component as referenced in the Solution is not valid.

### Support for LCN Schematic DX and image snapshot import

Control Architect now supports importing LCN Schematic DX and image snapshots into an LCN type configuration project.

#### **Control Conversion enhancements**

Control Conversion supports a new function <CPREF> that allows a user to specify connections to be added/replaced in an Excel file where the same Entity Name can occupy multiple data rows.

## **Installer enhancements**

Control Architect installer now preserves any changes made to the Mongod.cfg configuration file by the user when upgrading to a new release.

# What's New in Control Architect Release 4.30.5

#### Control Architect now targets the .Net Framework 4.6.2

Control Architect development is now targeting the .Net Framework 4.6.2.

#### **Defect Removal**

Many defects reported by others and found during testing and Control System project usage have been removed. View the detail release notes for specifics on defect fixes.

# **Control Architect now supports Mongo Server Authentication**

Mongo Server authentication of MongoDB databases is now supported. It is strongly recommended that users choose to use authentication to secure their data.

## Safety Manager FSC (SM) Support for integration of PKS CEE, FSC EE

Control Architect now supports the importing of FSC Excel FLD and Property definition files for an integrated Chart View rendering of CDA connections between C300 CEE entities and FSC FLD blocks. An FSC Control Module is generated during importing that shows a typical FLD drawing with all COM and FSC blocks embedded in the Control Module. FSC FLD Block to Block Peer connections are also visualized in the Chart views. The Peer Map view also shows any CDA peer between FSC Controllers and C300 CEE together with FSC Controller to FSC Controller Peer.

### **PKS Server configuration Import To MongoDB enhancements**

The PKS configuration import view now supports selecting multiple folders for HMIWeb graphic source files importing.

## **Chart View Printing/Saving**

Chart View printing/saving now includes a Control Module header that indicates the CM name, project name, and date.

# What's New in Control Architect Release 4.30.1

## **Control Architect is a full 64-bit Application**

Control Architect is now 64 bit and fully supports Windows 10 installation.

#### **Defect Removal**

Many defects reported by others and found during testing and Control System project usage have been removed. View the detail release notes for specifics on defect fixes.

#### Full support for PKS R430+ Chart View rendering

The Chart View rendering of R430+ configuration required significant changes to conform to the new R430 Visual model.

#### **MongoDB Document Server Technology**

Control Architect now fully supports integration with MongoDB Document Server technology.

Most configuration documents require importing in a MongoDB Server database through Control Architect. Control Architect no longer supports/maintains local file storage, but has moved to a Document Database that can reside on your local host or be scaled to a commercial Cloud or Server based hosting.

#### **MongoDB Server Maintenance View**

A new Server Maintenance view has been developed to allow users to administer their MongoDB Server database.

#### **Control Architect Reporting**

- Control Architect now supports an integrated Report View that supports report generation for supported view functions.
- Currently reporting is integrated with the HMIWeb Graphic View and the Control Architect Rule Authoring application.
- HMIWeb view supports many predefined reports to aid in project development and life cycle maintenance of HMI.

Reports can be exported to various document types including Microsoft Word, Adobe PDF, etc.

#### **Control Architect Rule Authoring Document Editor**

The CA Rule Author now supports an HTML editor for documenting Rule Sets that are executed in PolicyCop for validation of configuration.

#### **HMIWeb Graphics Capture**

A new HMIWeb graphic capture tool has been developed to allow users to document the WYSIWYG rendering from Honeywell HMIWeb Display Builder.

User can choose to copy this standalone executable to a Honeywell PKS Station or Server for execution or alternatively install HMIWeb Display Builder from the PKS DVD to your local development machine.

Graphic images can be captured as PNG or PDF file types.

#### Control Conversion supports Honeywell Win7 TPS Builder Application

Control Conversion activities now support Honeywell Win7 TPS Builder.

An API has been developed that allows the building of LCN based Control strategies by programmatically driving TPS Builder and its SqlServer database.

Importing LCN EB (Exception Build) configuration data using the Text File Conversion tool also support synchronizing with TPS Builder to import Control strategy containment (folders).

#### **Enhanced MRU Lists**

The existing MRU list that maintains most recently used Project names has been enhanced to support pinning of MRU items. This allows users to pin project items of importance that are not automatically rolled out of the recent list of project names when the MRU maximum limit of items is exceeded.

A new MRU list has also been implemented for Control Conversion solution files.

# **What's New in Control Architect Release 3.40**

# Control Architect has been tested and qualified for Windows 10 installation

Control Architect has been tested and qualified for Windows 10 installation.

## **Control Architect now supports 64-bit execution**

Users can choose to install the 32-bit or 64-bit versions. Users that require Honeywell TPS Builder Win7 integration need to remain on the 32-bit version.

### What's New in Control Architect Release 3.30

#### PKS R430+ Support

Control Architect fully supports PKS R430+ releases for building XML configuration files and for rendering the new Chart View visual model

## **Enhanced Control Conversion functions**

Entity references in HMIWeb graphic shapes are now visualized as invalid if they cannot be located in the Asset Model. Invalid entities are marked as "[INVALID] entityname" when data is pasted into Excel.

### New MRU (Most Recently Used) functions for Solution and Xml Project items

Enhanced the Most Recently Used (MRU) lists for both the Xml projects and Solution projects menu items. An MRU function has been added to the File-Open Solution menu to include a new drop-down menu button that displays the MRU list of items.

The MRU lists now provide the ability to pin an item to the Pinned category. Right mouse-click on an MRU menu item to display its Context menu and select the option "Toggle the Pin status for this item". Unpin items by again accessing this option to toggle its pinned status to unpinned.

#### **Enhanced HMI Web View**

Entity references in HMIWeb graphic shapes are now visualized as invalid if they cannot be located in the Asset Model. Invalid entities are marked as "[INVALID] entityname" when data is pasted into Excel.

Enhanced HMIWeb paste to Excel to include Parameter Update Period statistics for each graphic.

Modified order of data columns when pasting HMIWeb graphic data into Excel.

HMIWeb view data that is copied and pasted into Excel now has its Entity Reference split into 2 data columns Entity and Parameter.

#### **New Data View Container Grouping view**

A new Tree View has been developed for showing Module hierarchical relationships when grouping the Data View on its Container property.

There is additional support for the new Win7 Honeywell TPS Builder that allows synchronizing your TPS Builder projects to obtain Repository (database) and Project information for creating the TPSB Equipment Modules as XML Configuration files. This synchronization process occurs while converting your exported Exception Build files to PKS XML configuration standard.

#### **What's New in Control Architect Release 3.20**

#### A new Manifest viewer has been added to the Configuration views

The Manifest view allows the user to see the configuration sources that were used in generating the Asset Model. For mixed LCN/SCADA/ PKS projects you would expect to see a reference to the source used for the LCN EB file data, the SCADA pnt & hdw file data, and the exported Control Builder XML configuration files.

## **Support for OPCI references**

The Manifest view allows you to specify the OPCI data files copied from the PKS Server. When the configuration project is parsed, these OPCI files provide additional connection references that are visually shown as NonCEE type references in the Chart View.

## Support for Importing Strategy data into TPS Builder

Control Architect now supports direct importing of Strategies into TPS Builder. New Control Conversion activities were added to support this conversion type. The Control Property Import Wizard and the Control Conversion Template view now support additional UI controls for specifying the SQL Server connection to the TPS Builder Control Library and a Repository (Database).

## **Support for printing Chart View to PDF**

The Chart View now supports printing to PDF type documents. You can also copy the current Chart View to the Windows Clipboard as a bitmap type image for pasting into application such as Paint and Excel.

## What's New in Control Architect Release 3.10

## Support for duplicating items from within the Data View

Selected items in the Configuration views Data View tab can now be duplicated to quickly build PKS XML Configuration files

## Support for NonCEE type references

External references to NonCEE types such as SCADA and LCN are visually shown using a user defined color and a Watermark applied to the Function Block that makes use of the NonCEE reference.

#### **What's New in Control Architect Release 2.30**

## Support for identifying delved layers in Control Graph

When delving into Control Strategy relationships using Control Graph, the new exposed Vertex nodes as a result of delving into a particular entity will now show a colored border to identity its layer or group of entities that were exposed

# **Support for Node statistics**

The configuration views now provide a Tab named "Stats" that when clicked will analyze and show various statistics associated with the configuration project such as Memory Units, Process Units, block or device usage. A visual bar graph shows the usage stats on a device or block basis while bar graphs show the MU and PU stats for the project nodes.

Special support was added for LCN xPM type nodes such that Control Processor Point type summaries are shown as a bar graph that includes the used range, spare slot ranges, maximum available slots, etc. This data along with IOM Hardware slot usage can be copied to the Windows Clipboard for pasting into Excel.

#### **New Log Viewer**

A log viewer is now available for helping to troubleshoot and isolate defects and anomalies.

#### Support for PCDI Master Block peer viewing in the Peer Map view

PCDI Master Blocks are now shown in the Peer Map view.

#### **What's New in Control Architect Release 2.20**

#### Support for building LCN and SCADA point type Configuration through the Control Conversion functions

A new Workflow activity has been added that supports building LCN Exception Build (EB) files and PKS SCADA PNT text files when executed with a Data Collection.

#### Support for viewing xPM Logic and Device Control Gate logic in a visual Block view

When viewing LCN xPM type entities in the Chart View Logic and DevCtl point types will show an Adorner button that invokes a view that contains the visual Block configuration for the configured gates and inputs.

#### **Support for multiple tabbed Chart views and Configuration projects**

You can now view multiple Chart Views as tabbed views including the ability to position the views. You can also view side-by-side multiple configuration projects.

#### Support for enhanced filter expressions

Filter expressions now support the use of the bracket characters for specifying a range of values.

# What's New in Control Architect Release 2.10

## Support for LCN and SCADA point type Configuration has been added to Control Architect

A new conversion tool has been developed that converts LCN Honeywell Exception Build (EB) and SCADA (PNT) exported entities to XML Configuration file types.

Control Architect supports the conversion of other non-PKS CEE type configuration such as Honeywell Exception Build (EB) and SCADA Quick Builder to a compliant PKS Configuration standard. Control Architect uses the PKS CEE Configuration schema as the standard for all configuration views and tools. This Text File conversion <u>function</u> allows you to convert other non-PKS CEE configuration types to a compliant PKS Schema type.

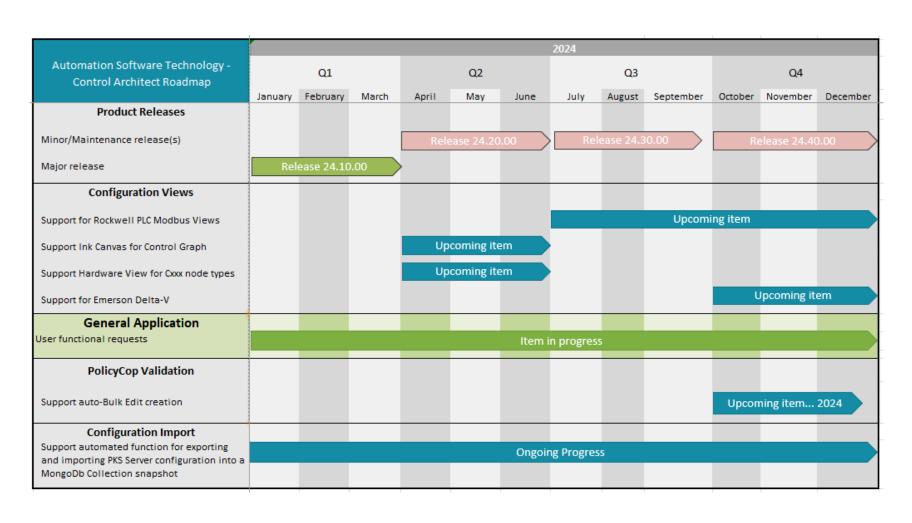
Entities parsed from Text files are generated as a simple Control Module with a single embedded Function Block where the embedded block represents the original parsed Entity. Block styles and attributes are configurable on a Template or Point type basis.

Other Control Architect functions such as Visual Peer Analysis, PolicyCop validation, Data Export, etc. work seamlessly with these other point types.

## **Control Conversion Workflow**

A new Workflow has been added to the <u>Workflow Manager</u> that allows you to quickly build the set of required Control Conversion files required to support a Bulk Build/Edit task.

# **Product Roadmap**



# **Getting Started**

# **System requirements**

Control Architect's recommended system configuration includes:

- Microsoft .Net 4.8.1 with latest Service packs installed
- Windows 10 or Windows 11 64 bit processor technology
- Minimum 8 Core Processor; minimum Intel Core i7-4820K
- Minimum 16GB of RAM
- 100GB of free disk space; Preferably Solid State Disk (SSD) technology
- 1280x1024 screen resolution or higher
- Minimum 2 MB on board Video memory higher end video processors are recommended as the UI technology used in Control Architect prefers hardware acceleration.

Typically Laptop or Desktop systems that are recommended for CAD type applications will provide adequate technology and performance needs including:

- Microsoft Surface
- Dell Work stations
- Dell Alienware

# **Getting help**

This help file can either be viewed on-line or off-line when installed with Control Architect. You can obtain the latest version as well as other formats of this help file on line at <a href="mailto:online-help">online-help</a>

#### **Off-line access**

The off-line help file is part of the Control Architect installation. To launch it, either press the F1 key or click the help button located on the Ribbon Toolbar **Help** tab.



### **On-line access**

To access and view the most recent Control Architect's help file on-line, launch a web browser to the following URL: <a href="http://www.autosofttech.com//Portals/0/On-Line-Help/ControlArchitect/index.htm">http://www.autosofttech.com//Portals/0/On-Line-Help/ControlArchitect/index.htm</a>

## Printing the help file

Alternatively, you can download and print a PDF or Word version of Control Architect's documentation from the following URL: <a href="http://www.autosofttech.com/downloads/ControlArchitect/ControlArchitect.doc">http://www.autosofttech.com/downloads/ControlArchitect/ControlArchitect.doc</a>

# **Control Architect licensing**

The Control Architect installation requires a license once installed. Contact <a href="mailto:support@autosofttech.com">support@autosofttech.com</a> to obtain purchase information and trial licenses.

When a purchase has been made, AST will email a license key that needs to be installed and authenticated.

- 1. Choose Help-Application License menu item from the ribbon main menu.
- 2. A licensing dialog window will appear that allows installing, removing and saving license keys.
- 3. Enter the license key obtained from AST and click the **Install** button.
- 4. The application will connect with the AST Authentication Web Service to authenticate the license key. This can take several seconds during which the status flashes "Authenticating".
- 5. If Authentication Web Service fails to connect or authenticate, then click the button to save the license to a file and send this file to <a href="mailto:support@autosofttech.com">support@autosofttech.com</a> for manual authentication.
- 6. If manual authentication was required, you will load the returned authenticated license using the Load License button.

Once a license has been installed it can be saved to an external file using the **Save License** button. This enables you to backup licenses to enable re-installation (i.e. after an OS reinstall) without having to re-authenticate the license.

# **Unloading a License Key**

If you choose to unload an authenticated license, the license key will automatically be deauthenticated and become invalid. You will have to contact AST to obtain another license key for loading.

# **How to purchase Control Architect**

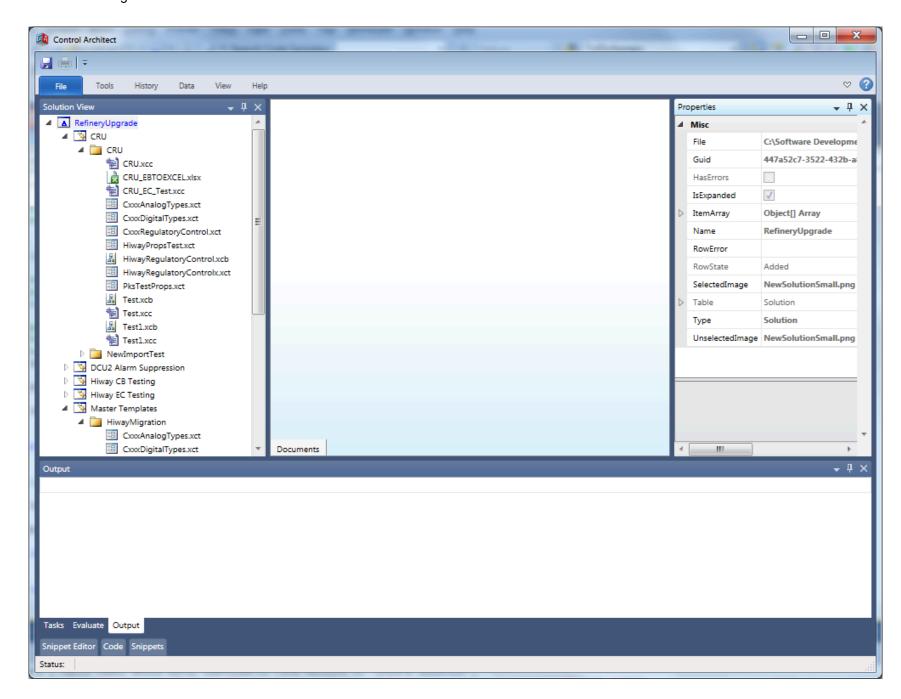
Control Architect licenses can be obtained from AST. Contact AST at <a href="mailto:support@AutoSoftTech.com">support@AutoSoftTech.com</a>. As soon as the transaction is complete, you will receive a license key for the application through Email. See the <a href="mailto:licensing">license</a> topic for more information regarding installing license keys.

# **Common Tips & Tricks**

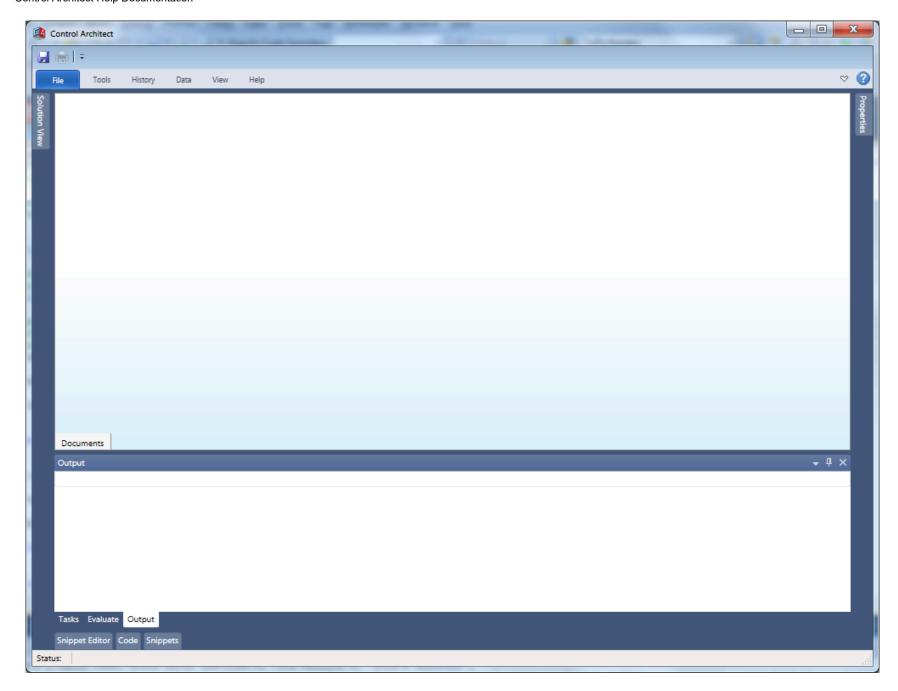
Control Architect is developed using Microsoft Windows Presentation Foundation (WPF) Technology that provides a richer User Interface (UI). There are common functions associated with a number of view controls that can be executed for example by using key stroke combinations.

# **Application Dock/Undock Docking Windows**

• You can undock and dock window views that support docking. The Application main window below shows the Solution View as a window docked to the left of the main application view, the Properties View docked to the right of the main application view, and the Output View docked to the bottom of the main application view. These docked views contain a set of docking controls located in the top-right of the their view that gives you control of their dock status. The Pin button image currently shows pinned for these 3 docked views. Clicking the Pin button will change the docked status for a view to undocked and it will become visible as an undocked tab.

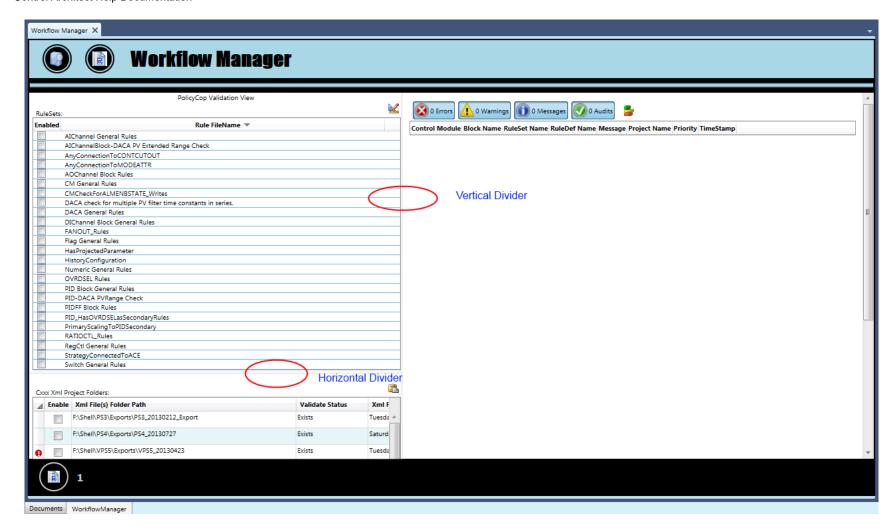


This same application view now shows the status when we undocked the Solution View and Properties View windows. Hovering your mouse cursor on the named View tab will float the window back into its normally docked view as long as the mouse cursor is maintained within the view. As the mouse cursor moves out of the view, its status will again become hidden. To dock the window to its normal position, click the Pin button while the view is focused with the mouse cursor to pin its status to docked.



# **Window View Dividers**

Many views contain horizontal and/or vertical dividers that can be dragged using the left mouse key to change view pane sizes. The Workflow Manager PolicyCop View below contains both a vertical and horizontal view divider that can be adjusted to give certain panes more or less view space. The dividers are shown as a light gray foreground color that when hovered over with the mouse cursor will change the mouse cursor to double arrows that indicates that the divider position can be changed by holding the left mouse key down and dragging the divider to a new position. In the view image below a vertical divider exists between the Rule Sets ListView pane and the PolicyCop validation results ListView pane. Another horizontal divider exists between the Rule Sets ListView pane and the XML Projects DataGrid View.



# **Code/Script Editor**

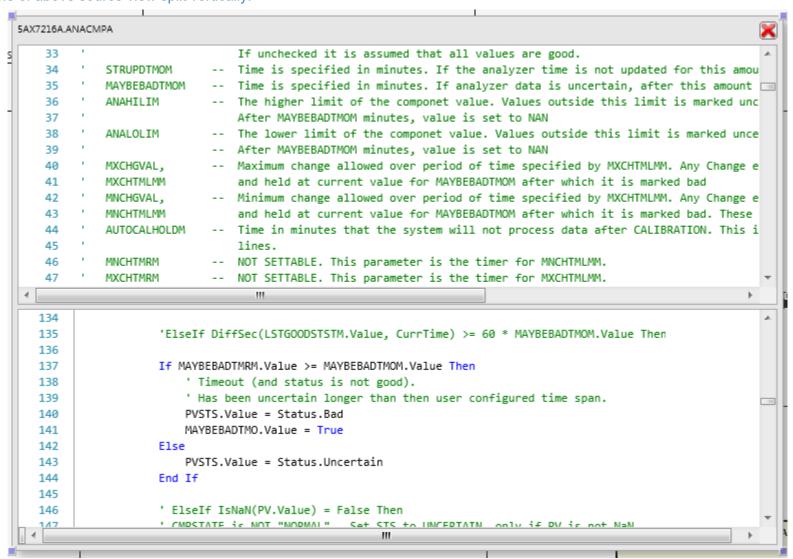
- Pressing the key combination Ctrl+A will select all text in the active view.
- Pressing the key combination Ctrl+Z will undo the last edit change.
- Pressing the key combination Ctrl+Y will redo that last undo operation.
- Pressing the key combination Ctrl+I will start an incremental search operation. Press Ctrl+I or Ctrl+Shift+I to navigate between search text
- Holding the Control (Ctrl) key while scrolling with the mouse middle scroll wheel will zoom in/out the view.
- Some code views support drag-n-drop such as dragging a code snippet to a Control Binding Template binding expression.
- Code view source can be copied and pasted as an RTF type to Microsoft Word to preserve syntax formatting such as color, etc.
- Code views support splitting the view surface either horizontally or vertically or both.

```
5AX7216A.ANACMPA
       1
              Program:
                           ANACMP
V٩
       2
              Author:
                           Pohheng Teoh (translating from CL)
                         5.02
       3
              Revision:
       4
              Last Update: 2006-07-31
                                                                                           Vertical Splitter Handle
       5
                           Based on existing Shell CL AMANACMP.CL
           ' Updates:
       6
                           Giri Mani, Honeywell. Changes made per discussion with Bob Stanton, Juan Cuellar
           ' Update Date: 6/17/09
       7
       8
       9
              This program is used to check the status of a single analyzer component.
      10
      11
              This program optionally provides checks for high, low, rapidly changing,
      12
              and dead component signal.
      13
              This program optionally integrates analyzer stream status (trouble,
      14
      15
              calibration) into the component checks.
      16
      17
           'Imports for the standard Honeywell supplied namespaces
      18
          □Imports Honeywell.CAB.SysCommon
      19
           Imports Honeywell.CAB.SysBase
      20
           Imports Honeywell.CAB.BlockBase
      21
           Imports Honeywell.CAB.Math
           Imports Honeywell.CAB.BlockMap
      22
      23
            'Imports for the Math namespace
      24
      25
          □Imports System.Math
           Imports System.Double

    Horizontal Splitter Handle

      27
      28
                  CDP Parameters
      29
```

## Example of above source view split vertically:



## **List Views**

- Pressing the key combination Ctrl+A will select all items in the active list view.
- Holding the Control (Ctrl) key while left mouse-clicking items will allow you to select multiple items.
- Holding the Shift key while left mouse-clicking item will select all items with the previously selected item and item clicked while holding the Shift key.
- You can mouse drag-select multiple items using your mouse by first holding the [Shift] key, then left mouse-click on the first item and drag the mouse to select multiple items.
- All list type views support copy to the Windows clipboard and pasting to Microsoft Excel using the key combination Ctrl+C to copy and Ctrl+V to paste.
- Some list views support left mouse-double clicking to invoke an action such as navigation.
- Pressing the key combination Ctrl+Shift+Plus where the Plus part denotes the '+' key will auto size all column widths to the displayed data values.

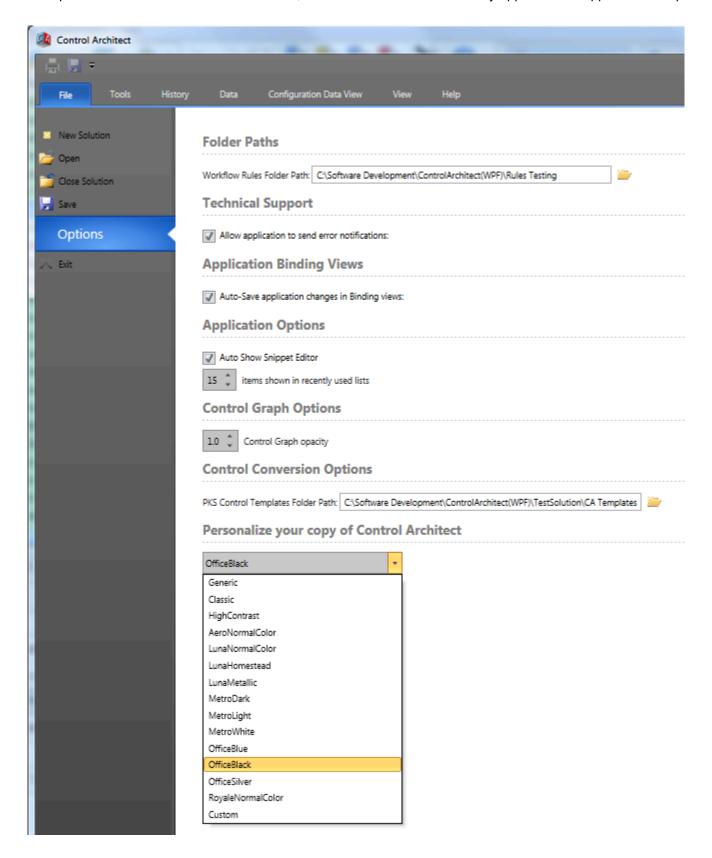
## **DataGrid Views**

- Pressing the key combination Ctrl+A will select all items in the active data grid view if the it support multiple selection. Not all data grids support multiple selection.
- Holding the Control (Ctrl) key while left mouse-clicking items will allow you to select multiple items where multiple selection is allowed.
- Holding the Shift key while left mouse-clicking item will select all items with the previously selected item and item clicked while holding the Shift key where multiple selection is allowed.
- All data grid type views support copy to the Windows clipboard and pasting to Microsoft Excel using the key combination Ctrl+C to copy and Ctrl+V to paste.
- Some data grid views support hyperlink navigation. Some data grid cells will contain a hyperlink text that can be left mouse-clicked to navigate to the item.
- Some data grid views support left mouse-double clicking on an item to expand the selected data grid row to expose more detail. This is typical for Control Conversion type template views.

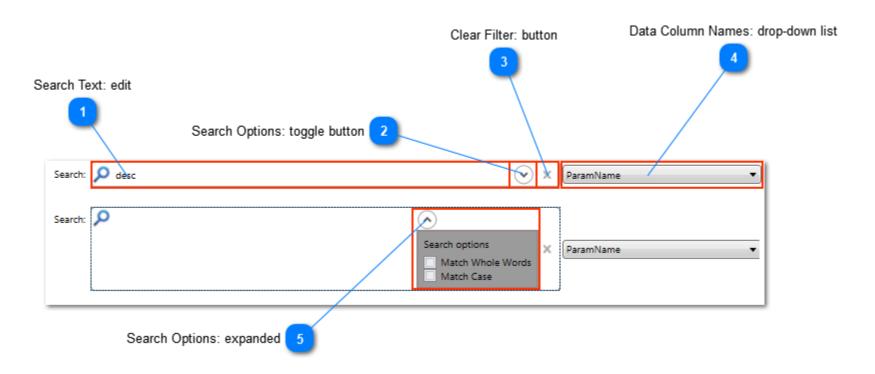
# **Application Themes**

Users can chose to apply a theme to their installed copy of Control Architect, which applies a predefined set of styles that affect the visual coloring used for various element types. Each user account can choose to apply their own theme type.

Control Architect supports a number of predefined themes that can be selected by accessing the Ribbon Toolbar *File* menu item and selecting the *Options* menu. When a theme is selected, its visual effects are immediately applied to the application components.



# **List View Filtering**



Generally a ListView type view element supports advanced filtering expressions using the wild card character set "!, \*, ?".

Using these special filtering characters allows you to filter the data to a particular subset.

## **Examples**

Wildcards are represented by "\*" or "?", where "\*" matches multiple characters and "?" matches only a single character. The special character "!" negates the result of the text search. You can also use the "|" pipe character to form a logical OR condition. For example, let's assume the DataView has the following items.



#### Note:

Character range search is also supported using the bracket grouping as in "2F[CF]" where the range group "[CF]" specifies to find either of the characters 'C' or 'F' in the 3rd character position. The characters specified in the range are not case sensitive.

2FC100 2FC200 2LC120 2FY200FC LOG2345MP LOG2345 MP2345LOG

The following filters will produce the following results.

#### Command

Filter Result

FC Will return 2FC100,2FC200,2FY200FC

!FC Will return LOG2345MP,LOG2345,MP2345LOG

!LOG Will return 2FC100,2FC200,2FY200FC

\*MP Will return LOG2345MP

MP\*LOG Will return MP2345LOG

?FC??? Will return 2FC100,2FC200

LOG\* Will return LOG2345MP,LOG2345

!LOG\* Will return 2FC100,2FC200,2FY200FC,MP2345LOG

?F[CF] Will return 2FC100, 2FF100, 9FF100, but will not return 2FT100

?L|?F Will return any tags that have either the character 'L' OR 'F' in the second character position {2FC100,2FC200,2LC120}



#### **Notes**

- When using wildcards, it is possible to use multiple wildcards within the same token.
- The \* character matches one or more characters in the position of the string value being searched.
- The ! negation character can only be used as the first character in your filter text value.



Specify the filter text to apply to the selected data column.

Search Options: toggle button

Click this header button to toggle the expanded/collapsed state for the search options. After choosing the required options, click the toggle button to collapse the options to allow specifying the filter text.

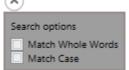
Clear Filter: button

Click this button to clear the current filter results in the attached ListView.

Data Column Names: drop-down list

This drop-down list will contain a list of column names available for filtering on the data in its attached ListView element.

Search Options: expanded



Search options are available by clicking the header label to toggle its expanded state.

## **Common Control Architect Tasks**

Control Architect is designed to work with Honeywell configuration files that have been exported from the Control System. Common file types that you will need to acquire when working with Control Architect are:

# **Cxxx/ACE** Configuration

- EPKS Server Cxxx/ACE XML Configuration files files are exported from the Server ERDB
- HMIWeb Process Graphic files files are typically located in the Abstract\00. Graphics folder

# **SCADA Configuration**

Quick Builder exported PNT & HDW Configuration files - files are exported from the Quick Builder application

# **LCN Configuration**

LCN exported Exception Build (EB) Configuration files - files are created using LCN PED Print Entities functions

# **Honeywell Hiway Analysis**

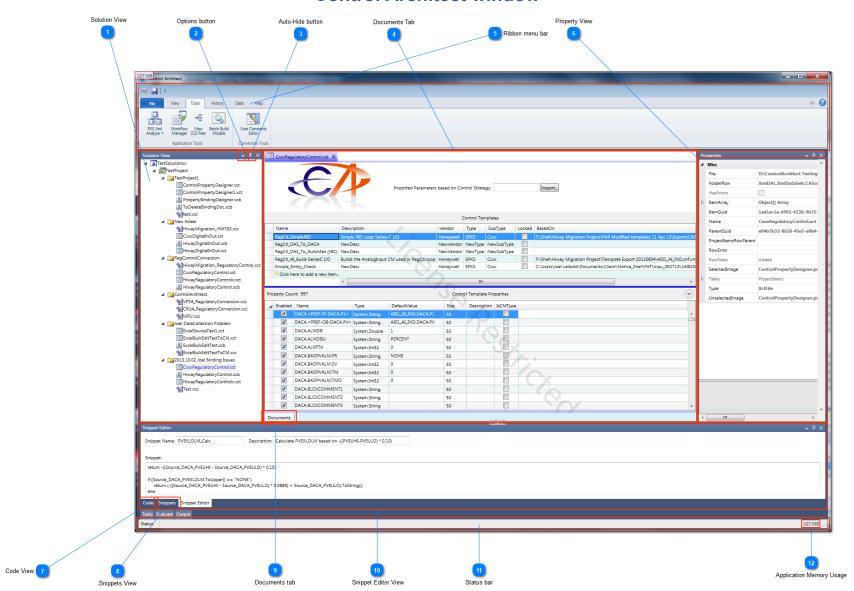
- Hiway and UCN Exception Build (EB) files files exported from the LCN using Native Window Engineering functions
- Control Language (CL) files AMCL/PMCL/MCCL files are copied from a LCN History Module

### **Common Control Architect Tasks**

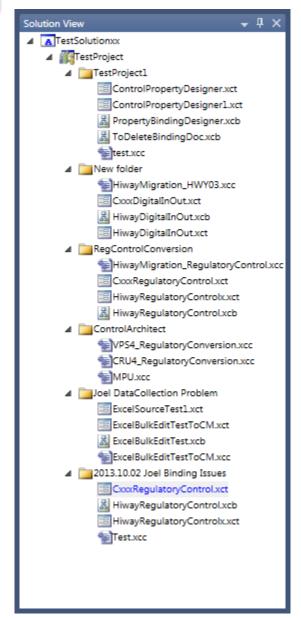
- Getting Started Using Control Architect for a PKS Cxxx/ACE Project
- Getting Started Using PolicyCop Validation

# Overview of the user interface

#### **Control Architect window**



Solution View



The Solution View contains project files used to support Control Conversion tasks.



# **Options button**



This button allows you to float and dock Tool an Document type windows.

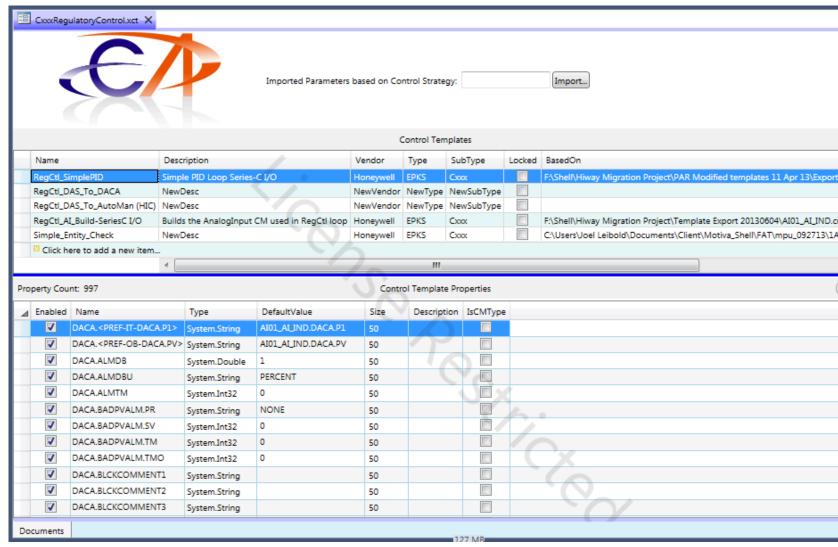
3

This button allows you to pin open a Tool window.

4

## **Documents Tab**

**Auto-Hide button** 



Documents opened from the Solution View appear in the [Documents] tab.

5

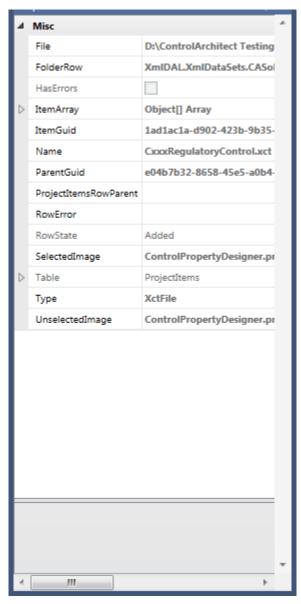
## Ribbon menu bar



The Ribbon menu bar provides access to tools and functions found in Control Architect.

6

# **Property View**



The Property View shows details associated with Solution View file type items.

Code View



The Code View displays C# code that is used in Control Architect features such as Control Conversion.

Snippets View



The Snippets View displays a list of user created C# code snippets that can be reused in Control Conversion Control Binding expressions.

Documents tab



The [Documents] tab contains all documents that are opened from the Solution View.

Snippet Editor View



The Snippet Editor View allows you to edit user created code snippets.

Status bar

The Status bar displays information related to application execution. Notifications including task status, errors, warning, etc. are displayed he

Application Memory Usage

This is the applications private working set memory usage that is updated at 5 second intervals.

If this value is displaying 0, then follow this procedure to rebuild the performance counters for your system as they are corrupted.

Copyright © 2024 by Automation Software Technology, Inc. All Rights Reserved.

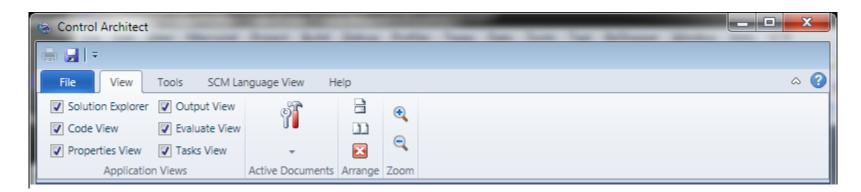
Click Start, type **cmd**, right click cmd.exe, and select Run as administrator. At the prompt, type *lodctr /r* and press ENTER. This will rebuild the performance counters for your system.

If this fails with the result message Error: Unable to rebuild performance counter setting from system backup store, error code is 2, then try this in the Command window:

C:\windows\SysWOW64> lodctr /r

You should get a results message Info: Successfully rebuilt performance counter setting from system backup store

#### **Ribbon Tabs**

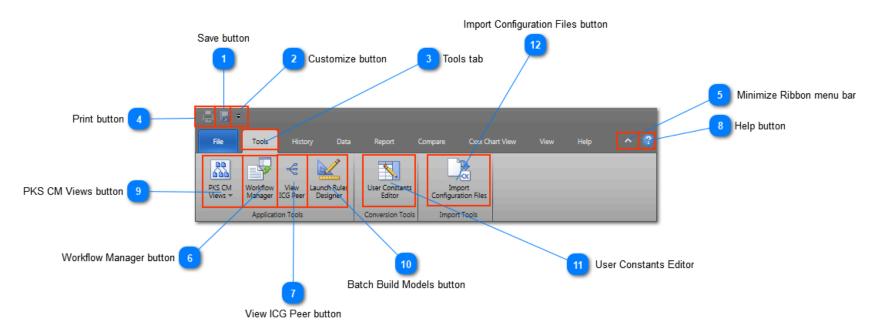


#### **Presentation**

Control Architect's ribbon tabs are located at the top of the main window and provide all the features available within Control Architect in a categorized fashion. The ribbon tabs parts are:

- 1. The main tabs They are always visible and are used for the most important actions
- 2. The tabs groups When a tab is selected, it will display actions grouped by similar purpose
- 3. Contextual tabs These tabs are only shown when needed. For example, the "SCM Language View" tab is only visible when an SCM Control Module is currently open in a Chart View.

#### **Tools menu**







Click this button to save changes to document views such as the Control Conversion document views.

Customize button



Click this dropdown button to change what buttons are visible in this mini-Toolbar section.

🗻 Tools tab



This tab contains application tools.

Print button



This button provides access to print functions for document views that support printing.

Minimize Ribbon menu bar



Click this button to toggle the minimized state of the Ribbon menu bar.

Workflow Manager button



Click this application tool button to invoke the Workflow Manager.

View ICG Peer button



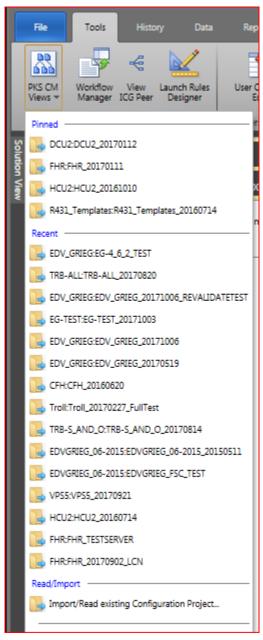
Click this application tool button to invoke the ICG Peer analysis view.

Help button



Click this button to view application Help.

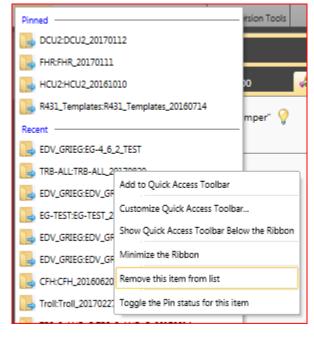
PKS CM Views button



Click this application tool button to view a list of Most Recently Used PKS Projects or browse for an existing PKS XML Project. Click the **Browse for PKS XML Project** to browse for a folder path that contains exported PKS XML configuration files. If this is the first time a PKS XML project is being viewed, you must first click the <u>Parse</u> button located in the *Data View* tab to parse the exported XML configuration files to a Configuration Asset Model. This Asset Model is used by Control Architect to navigate your PKS configuration project entities and relationships.

When a new XML folder path is browsed, the folder path is saved at the top of a MRU list. This MRU (Most Recently Used) list maintains a <u>user specified number</u> of last used items.

You can choose to remove an existing item from the MRU list or toggle the Pinned status of an item by right mouse-clicking to display a context menu and choosing to remove the item or toggle the pinned status.



# Batch Build Models button



Click this application tool button to invoke the Batch Build Models function.



### **User Constants Editor**



Click this application tool button to invoke the User Constants Editor.

12

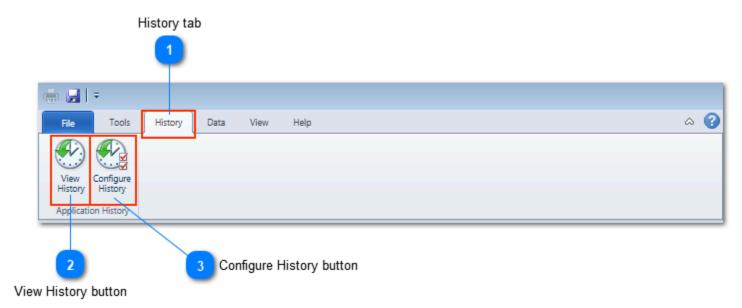
# **Import Configuration Files button**



Click this application tool button to invoke the Import tool. This tool allows you to convert Honeywell Exception Build (EB), Honeywell FSC (SM), and Quick Builder SCADA export files to XML configuration files that are an abstraction of a PKS XML Configuration schema. LCN and SCADA entities are generated as a simple Control Module with a single embedded Function Block where the embedded Function Block represents the original entity. The generated Function Block contains visible block parameters and pin connections. Connection relationships (including CL entity/parameter and Equipment List Template usage), Control Graph (Control Strategy View), Visual Peer analysis, and Policy Cop validation are supported with the new converted types.

FSC Excel export files are imported by creating a Control Module that represents the FSC FLD with embedded FSC and COM Input/ Output blocks that show connection relationships between HMIWeb graphics, other CEE CDA reference blocks and FSC FLD-to-FLD Controller Peer references.

# **History menu**





This tab contains the History functions.

View History button



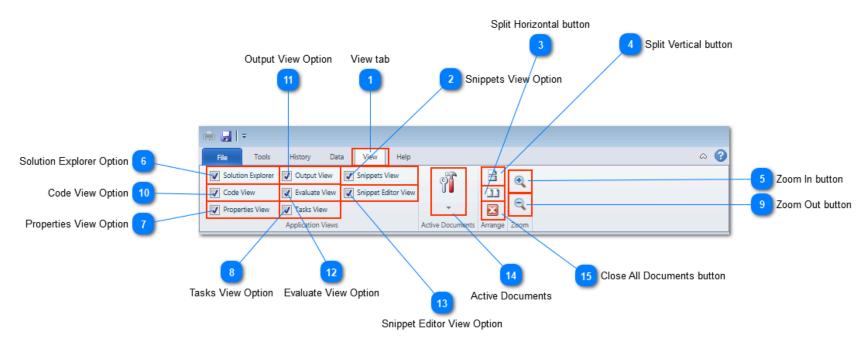
Click this button to view history associated with Control Architect functions that support history collection. Currently the only application function that maintains history is the Control Conversion Bulk Build/Edit.

Configure History button



Click this button to configure history file locations and deletion of existing History data.

#### View menu



1 View tab

This tab contains functions associated with the application views such as view visibility, active document list, arranging windows and zooming.

2 Snippets View Option

Snippets View

Check this option to view the Snippets view.

3 Split Horizontal button

Click this button to split the current active document into a new horizontal tab group.

Split Vertical button

33

Click this button to split the current active document into a new vertical tab group.

Zoom In button

•

Click this button to zoom in the active view. Note that not all views support zooming.

Solution Explorer Option

▼ Solution Explorer

Check this option to view the Solution Explorer view.

Properties View Option

▼ Properties View

Check this option to view the Properties view.

Tasks View Option

▼ Tasks View

Check this option to view the Tasks view.

Zoom Out button



Click this button to zoom out of the active view. Note that not all views support zooming.

Code View Option

Copyright © 2024 by Automation Software Technology, Inc. All Rights Reserved.



Check this option to view the Code view.

# Output View Option



Check this option to view the Output view.

# Evaluate View Option



Check this option to view the Evaluation view.

# Snippet Editor View Option



Check this option to view the Snippet Editor view.

# Active Documents



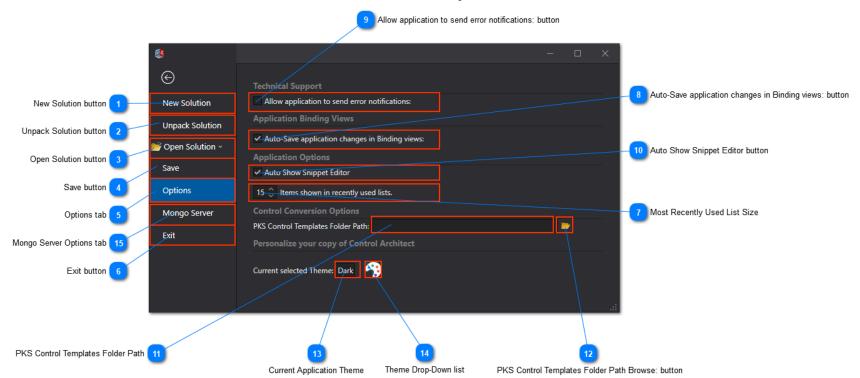
Click the dropdown button to view a list of opened document views. You can switch to a document view by selecting an item in this dropdown list.

# Close All Documents button



Click this button to close all opened document views.

# **File Menu-Options**



New Solution button

**New Solution** 

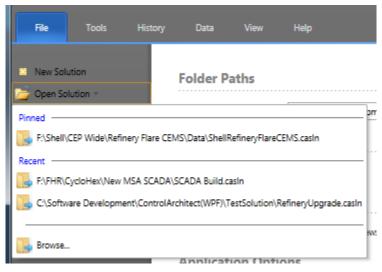
Click this button to create a new Solution.

Unpack Solution button

**Unpack Solution** 

Click this button to unpackage a Solution package file that was previously packaged using the Solution Package function.

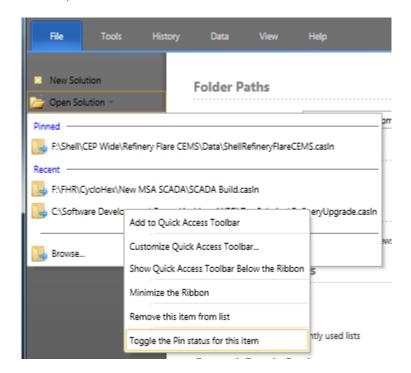
Open Solution button



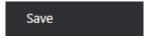
Click the Open Solution drop-down button to view a list of Most Recently Used Solution projects or browse for an existing Solution project. Click the **Browse item** to browse for an existing Solution project file \*.casln.

When a new Solution project file is browsed, the file path is saved at the top of a MRU list. This MRU (Most Recently Used) list maintains a <u>user specified number</u> of last used items.

You can choose to remove an existing item from the MRU list or toggle the Pinned status of an item by right mouse-clicking to display a context menu and choosing to remove the item or toggle the pinned status.



Save button



Click this button to save any changes in the active view.

Options tab



Access to other common application options.

Exit button



Click this button to exit the Control Architect application.

Most Recently Used List Size

15 👶 Items shown in recently used lists.

Specify the maximum number of MRU items to store where used throughout the application.

Auto-Save application changes in Binding views: button

Auto-Save application changes in Binding views:

Choose this option to auto-save changes instead of prompting the user to save the current changes when the application is required to proceed.

Allow application to send error notifications: button

Allow application to send error notifications:

Choose this option to automatically email error notifications encountered while using Control Architect to Automation Software Technology technical support group.

There is no additional information collected other than the technical details of the defect itself, which helps our support group to identify potential defects and their cause. When a defect notification is received, a representative from AST may contact the user to gain further insight into the issue.

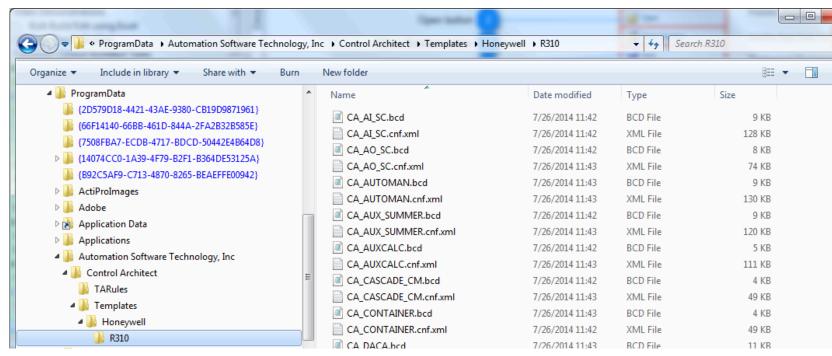
Auto Show Snippet Editor button

Auto Show Snippet Editor

Choose this option to automatically bring into focus the Snippet Editor view window when a code snippet item is selected in the Code Snippet view.

PKS Control Templates Folder Path

The location where your PKS Control Module templates or typicals are located. The Control Architect installation installs a set of Control Module templates or typicals are located at



# PKS Control Templates Folder Path Browse: button



Specify the location where your PKS Control Module Template files are located. The Control Conversion functions will look here to reference any PKS Templates that are required to support Control Conversion tasks.

# Current Application Theme



The current theme being applied to the application.

## Theme Drop-Down list



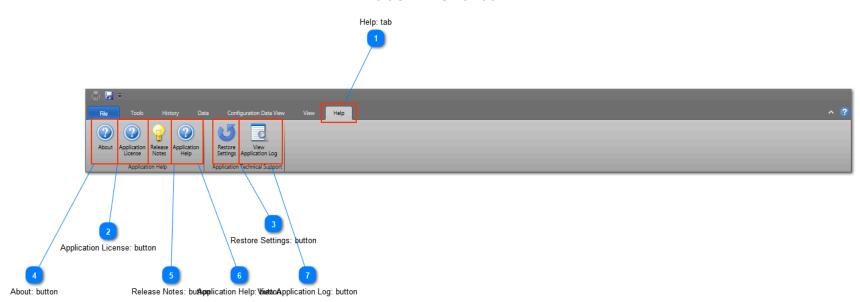
Select from the drop-down list a theme to apply to your application. You can choose from a list of pre-defined themes. You can also choose to use the option to synchronize the application theme to your current Windows 10 Theme, which will automatically switch the application theme if you change your System theme.

### Mongo Server Options tab



Click this tab to view/change Mongo Server options.

#### Ribbon menu bar



Help: tab

This Toolbar Tab allows you to access function related to the Help documentation, release notes, settings, license and application log.

Application License: button



Click this button to view and/or load a license to activate Control Architect.

Restore Settings: button



Click this button to restore the saved applications settings to their default values. When the application settings are restored, the application must be restarted.

About: button



Click this button to view the application About view. You can also view what options are currently licensed for the application.

Release Notes: button



Click this button to view the installed Release notes and History of changes for this application.

Application Help: button



Click this button to view the Help document for this application.

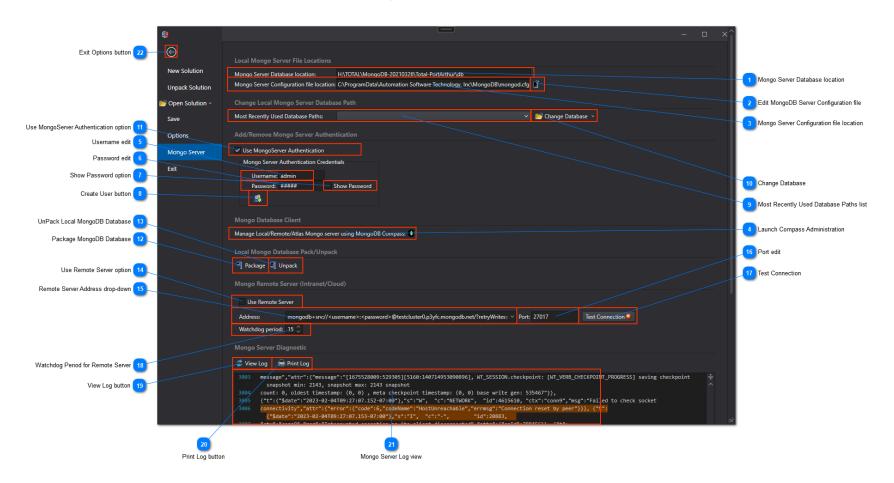
View Application Log: button



Copyright © 2024 by Automation Software Technology, Inc. All Rights Reserved.

Click this button to view the Application log. This log tracks all errors encountered when using the application and also tracks other important information that helps AST support in diagnosing failures. When exceptions or errors are displayed by the Application, you may be asked by AST support to provide this log for troubleshooting.

# **Mongo Server Options**



# Mongo Server Database location

Mongo Server Database location: H:\TOTAL\MongoDB-20210326\Total-PortArthur\db

Displays the folder path where the MongoDB Server database resides.

Users can choose to relocate this folder and its data contents by stopping the MongoDB Service, moving the folder and its contents, updating the Mongo Server Configuration file to point to the new folder path and restart the MongoDB Service.

MMCSS	1100	Multimedia Class Scheduler	Running	netsvcs
MongoDB	2392	MongoDB	Running	N/A
A II A II I		M	81 1	51/5

#### Edit MongoDB Server Configuration file



Click this button to open the configuration file in Notepad for editing. Note that any changes to this file will not be used by the MongoDB Server until its Service is restarted.

## Mongo Server Configuration file location

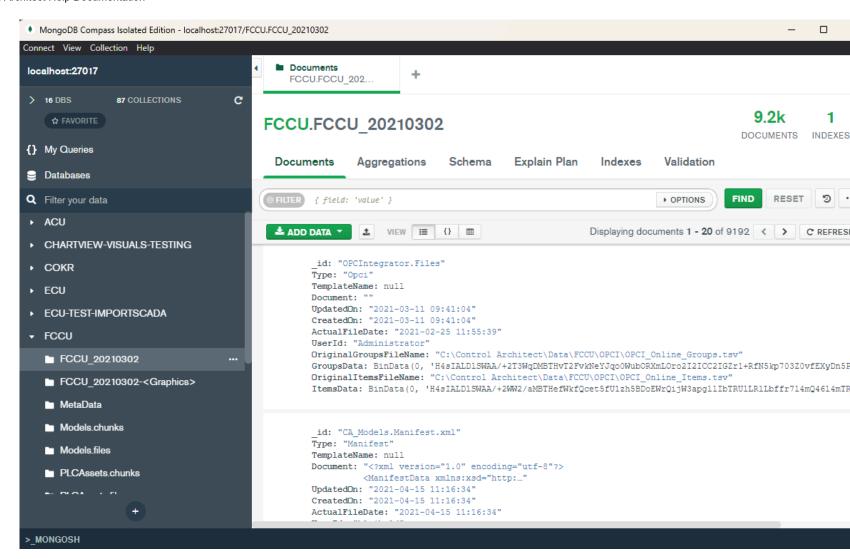
 $Mongo\ Server\ Configuration\ file\ location:\ C:\ Program\ Data\ Automation\ Software\ Technology,\ Inc\ Mongo\ DB\ mongod.cfg$ 

Displays the current file path where the MongoDB Server configuration file resides. Users should only have to edit this file when moving the installed location of the MongoDB Server database. Click the button next to this text box to open the configuration file in Notepad for editing. Note that any changes to this file will not be used by the MongoDB Server until its Service is restarted.

#### **Launch Compass Administration**

Manage Local/Remote/Atlas Mongo server using MongoDB Compass:

Click this button to launch the MongoDB Compass administration tool. This tool provides all the tools necessary to maintain your MongoDB Users should not use this tool to delete any Collections maintained by Control Architect as this could result in ghost references. Control Architect both Collections and Databases while assuring that referential integrity is maintained. This tool is installed for those users that have according with MongoDB database deployments.



Username edit

Username: admin

Specify a User name for the Mongo Server account.

Password edit

Password: ####

Specify a Password for the Mongo Server account. This Password is saved to your local settings using Microsoft System Security Cryptography API, which significantly reduces the chances of your account credentials being hacked. This Password is only valid for the user that is logged into the Windows account. Other users with valid login accounts will also have to establish their own Mongo Server authentication account.

Show Password option

Show Password

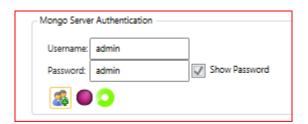
Choose this option to show the masked Password characters.

Create User button



Click this button to create a new user account with administration rights for the currently connected physical MongoDB database.

Clicking this button will start and stop the Mongo Server multiple times during the account creation process. The status of the Mongo Server will be displayed along with a progress indication while the new user account is being created. Wait until the progress indicator is no longer visible and the Mongo Server status LED is showing a "good" green status indication, which indicates successful completion.



If you manage multiple physical MongoDB databases in your file system, you are required to create authentication accounts for each database. Click the *Create User* button when connecting to a new physical MongoDB database to create the new user account.

When using the Mongo Compass application you will also be required to create a new connection that uses the same authentication credentials you specified here.



## **Most Recently Used Database Paths list**



Click this drop-down list to view a list of recently used physical MongoDB database paths that can be selected for attachment to your Local MongoDB Server. Select a physical database path from this list and then click the *Change Database* button to change the attached MongoDB server database.

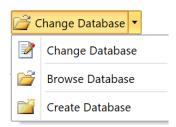


#### **Change Database**

#### 📂 Change Database 🔻

Click the drop-down button to view a list of options to execute against your Local MongoDB Server. Clicking the button itself will execute the default option to change the Local physical database path to the selection in the drop-down list.

Select a menu item from the drop-down list to execute.





These actions only apply to your local MongoDB Server installation and has no effect on a Remote Intranet or Enterprise Atlas Server installation.



### Use MongoServer Authentication option

✓ Use MongoServer Authentication

Choose this option to enable authentication for your local Mongo Server logon. Enterprise Atlas Mongo Cloud connections use these specified credentials to update the embedded "<admin> and and and and and and authentication. Remote Intranet Mongo Servers will also use the specified User Name and Password to authenticate logins.



# Package MongoDB Database

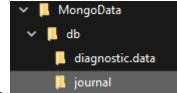
#### Package

Click this button to create a deployable Package file of your local or remote MongoDB physical database. You can choose what logical Databases are packaged. Typically you will use this function to copy/move a local Host Mongo database to another physical location or to create a periodic backup.

This is the recommended way to both upgrade and/or deploy a copy of a local database to another local or remote installation.

Users can optionally copy a physical MongoDB database by:

- 1. Stopping the Mongo Service
- 2. Copying the physical MongoDB database path to media
- 3. Copying the media copy to another physical database location and configuring the Server configuration file to reference this new database path
- 4. Delete the files located in the MongoDB database path "journal" folder to remove any transaction history



a.

5. Restart the Mongo Service on the new server installation

Example MongoDump.exe and MongoRestore.exe Command Prompt usage:

D:\Program Files\MongoDB\Server\bin>mongorestore /archive:"E:\TestDump\EDV\_GRIEG.EDV\_GRIEG\_20170519-%Graphics%.gz" /gzip 2018-05-20T14:43:54.040-0600 preparing collections to restore from

```
2018-05-20T14:43:54.064-0600 reading metadata for EDV_GRIEG.EDV_GRIEG_20170519-<br/>
\text{Graphics} from archive 'E:\TestDump \text{EDV_GRIEG_20170519-} \text{Graphics} gz' \text{restDump \text{CDV_GRIEG_20170519-} \text{Graphics} from archive 'E:\TestDump \text{EDV_GRIEG_20170519-} \text{Graphics} from archive 'E:\TestDump \text{EDV_GRIEG_20170519-} \text{Graphics} gz' \text{2018-05-20T14:43:57.006-0600} \text{EDV_GRIEG_EDV_GRIEG_20170519-} \text{Graphics} 358MB \text{2018-05-20T14:43:59.001-0600} \text{EDV_GRIEG_EDV_GRIEG_20170519-} \text{Graphics} 592MB \text{no indexes to restore} \text{2018-05-20T14:43:59.002-0600} \text{finished restoring EDV_GRIEG_EDV_GRIEG_20170519-} \text{Graphics} \text{(647 documents)} \text{done} \text{done} \text{done} \text{(647 documents)} \text{done} \text{done} \text{(647 documents)} \text{done} \text{done} \text{(647 documents)} \text{done} \text{(647 documents)} \text{done} \text{(648 documents)} \text{done} \text{(648 documents)} \text{done} \text{(649 documents)} \text{(649 docume
```



Special characters used in Database Collection naming such as '<' and '>' must be renamed as above using the '%' character to form valid archive file names when dumping the contents of a Database:Collection. This would typically apply to Collections that contain Process Graphic documents where the collection name is formed as DatabaseName:CollectionName\_TimeStamp-<Graphics>.

# 13

## **UnPack Local MongoDB Database**

Unpack

Click this button to unpack a previously packaged MongoDB database to your local Host or Remote Mongo Server. Users have the option to create a new physical Mongo database path while unpacking to a local Host Server only.



#### **Use Remote Server option**

Use Remote Server

Choose this option to connect to a Remote intranet Mongo Server or Enterprise Atlas Cloud installation.

Provide the IP Address for your remote intranet Mongo Server installation or a Connection string for your Enterprise Atlas Cloud installation in the <u>Address</u> field.

Provide a Port address for your remote intranet Mongo Server installation only.

If authentication is required, then provide your <u>UserName</u> and <u>Password</u> credentials and choose the <u>Use MongoServer Authentication option</u>.



### Remote Server Address drop-down

Address: mongodb+srv://<username>:<password>@testcluster0.p3yfc.mongodb.net/?retryWrites=

Provide the IP Address for your remote intranet Mongo Server installation or a Connection string for your Enterprise Atlas Cloud installation.

IT administrators will either provide you an IP address to connect to your Intranet server installation or provide a DNS host name that resolves to the IP address as specified in your system HOSTS file.

Connecting to an Enterprise Atlas Cloud installation will require a connection string that will be provided by your IT administrator that must be entered in this Address field.

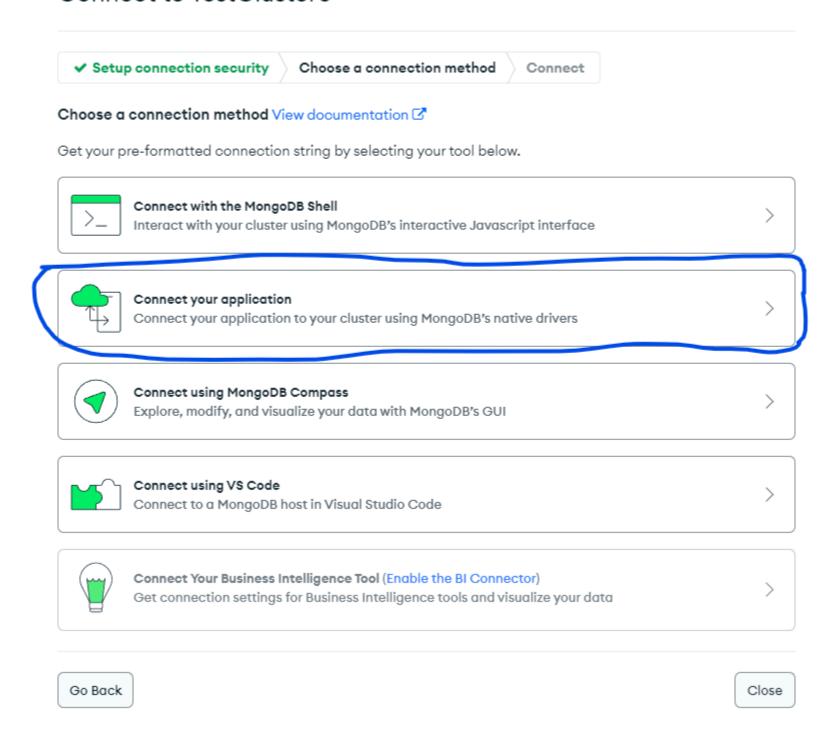
This is a typical connection string to use when connecting to a Mongo Atlas Cloud server:

```
mongodb+srv://<username>:<password>@testcluster0.p3yfc.mongodb.net/?
retryWrites=true&w=majority
```

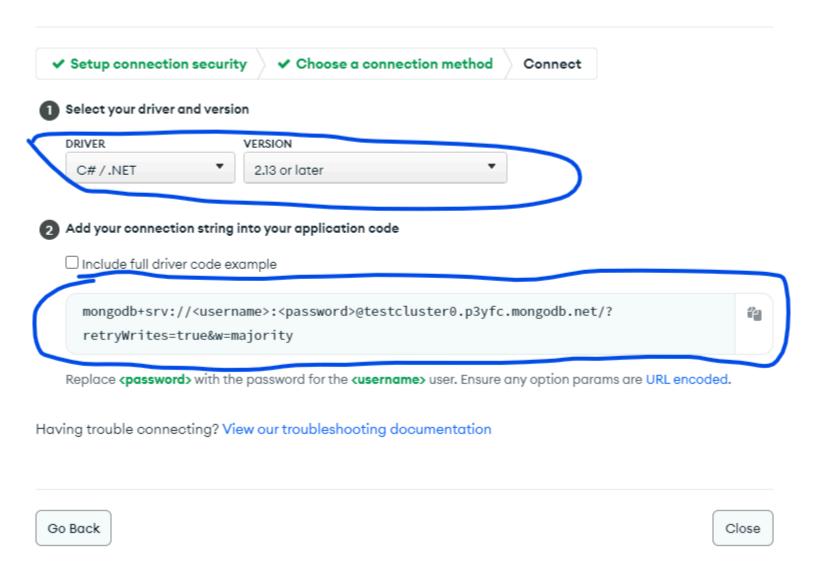
The connection string above begins with the format "mongodb+srv://UserName:Password" where UserName is automatically replaced by the Username text value and "<password>" is automatically replaced by the decrypted Password text value while connecting. Do not replace the text phrases <username> and <password> or an error will occur.

When connecting to a MongoDB Atlas cluster the required connection string is provided to you by accessing the "Connect" button on your Atlas home page and choosing to connect from an application using the driver: "C#/.NET" version: "2.13 or later".

# Connect to TestCluster0



# Connect to TestCluster0



# Port edit

Provide a Port address for your remote intranet Mongo Server installation only. The default port of 27017 can be used in most cases.

# Test Connection Test Connection

Click this button to validate your remote MongoDB Server connection. This will test your authentication credentials, connection string for Mongo Atlas deployments, and port usage for remote intranet deployments.

# Watchdog Period for Remote Server Watchdog period: 15 \$

Specify the Watchdog period in seconds as to how often to periodically poll the remote server to verify MongoDB Server connectivity.

# View Log button

荐 View Log

Click this button to view the current MongoDb Server Log content. The Log content is retrieved from the Mongo Server memory

If the Mongo Server is local and the Server has stopped, then the log content is retrieved by reading the physical Mongo database log file. If an attempt to retrieve the log for a remote server fails because the remote server is not reachable, then the physical log file must be retrieved from the Mongo Server's file system.

# Print Log button



Click this button to print the content of the Mongo Server Log view.

# Mongo Server Log view

Copyright © 2024 by Automation Software Technology, Inc. All Rights Reserved.

Click the *View Log* button to view the connected Mongo Server log that is cached in memory. If the Server is not reachable due to a bad connection or a failed MongoDB Service, then this function will attempt to retrieve the physical log file from the local Mongo Server connected database. The physical log file of a remote Mongo Server cannot be retrieved using this function.

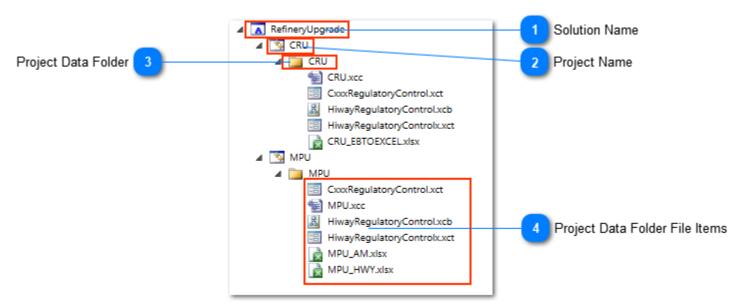


# **Exit Options button**



Click this button to return back to the main application view.

#### **Solution View**



# Solution Name

RefineryUpgrade

When creating a new Solution, you specify this name. The solution contains all your projects, project data folders, and project file items.

• To add a new Project right click on the Solution item to display a context menu and choose the menu item Add-New Project.

# Project Name

😘 CRU

This is a Solution project that contains project data folders and their project file items.

- To add a new Project data folder right click on the Project item to display a context menu and choose the menu item *Add-New Folder*. You will be prompted to browse for a file system folder, which will become the default project data folder name.
- To rename the Project name right click on the Project item to display a context menu and choose the menu item *Rename*.

# Project Data Folder

CRU

This a Solution Project data folder that is used to organize related data source items such as Control Architect template instances, Excel data sources, Word documents, etc.

- To add a new Control Architect Template item right click on the Folder item to display a context menu and choose the menu item Add-New Item...
- To add an existing file item such as a Control Architect Template instance, Excel file, Word file, etc., right click on the Folder item to display a context menu and choose the menu item *Add-Existing Item...*
- To rename the Folder name right click on the Folder item to display a context menu and choose the menu item Rename.

#### Project Data Folder File Items

CxxRegulatoryControl.xct

MPU.xcc

HiwayRegulatoryControl.xcb

HiwayRegulatoryControlxxct

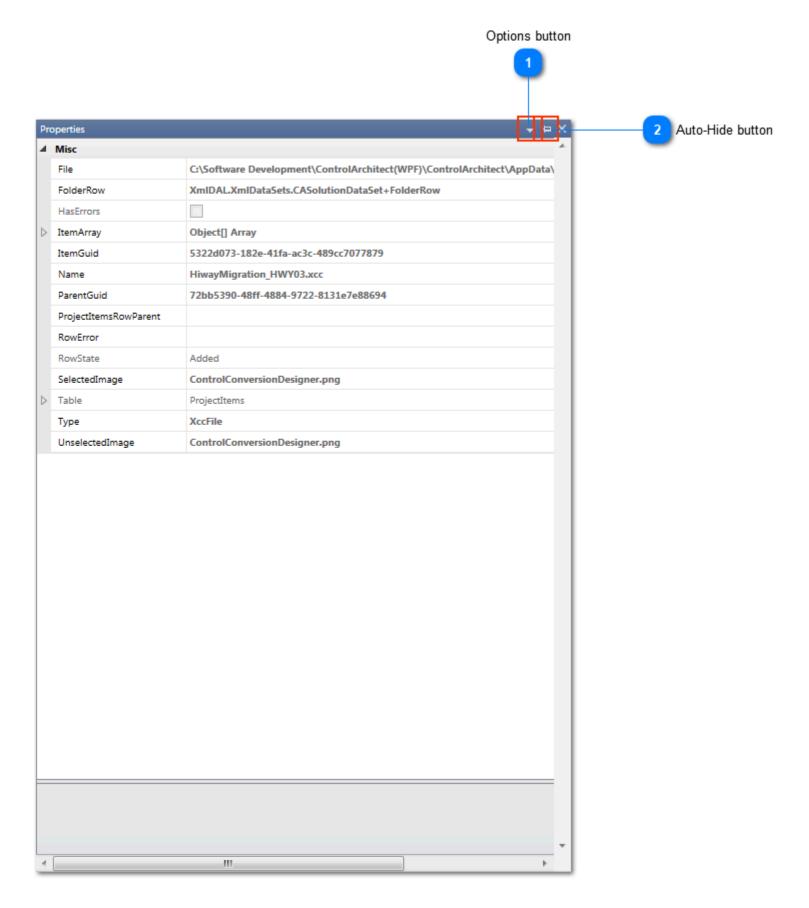
MPU\_AM.xlsx

MPU\_HWY.xlsx

These are file items that are contained in a Project data folder. Typically you will organize and add file items that are related to the project task such as a Control Conversion from a Honeywell Hiway based control system to an EPKS Experion C300 based control system.

# **Properties view**

The Properties view provides detailed file based information for files contained in the <u>Solution view</u>. The data that is presented in this view reflects the current item that is selected in the Solution view.







Click this Dropdown button for docking options.

# 2 Auto-Hide button

Click this button to Auto-Hide or pin the docking window.

#### **Code view**

The Code view provides a read-only view for code functions within Control Architect. It provides a larger view for reading code.

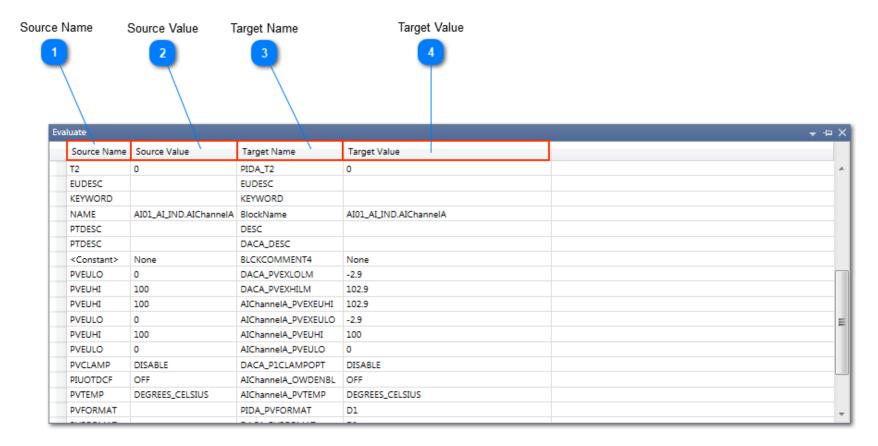
When developing Control Binding expressions used in Control conversion tasks, the option to compile and evaluate will generate a Control class that contains all the Control Binding expression code snippets. This generated Control class will automatically be visible in this view. Any errors emitted from the code compiler will reference source code line numbers that can be viewed to help diagnose the source of the error.

```
Code
                                141
                                142
                                               // Converted Target Control Property
                                               private System.String DACA PVFORMAT;
                                143
                                               [PropertyBinderAttribute("PVFORMAT", "DACA.PVFORMAT")]
                                144
                                145
                                               [PropertyRoundingAttribute("2", false)]
                                               [PropertyDefaultValueAttribute("")]
                                146
                                147
                                              public System.String DACA_PVFORMAT
                                148
                                               {
                                                   get
                                149
                                150
                                                   {
                                151
                                                       // TODO: modify this get() accessor method to accomplish
                                152
                                                       // Source Control Property value conversion
                                                       if(Source_DACA_PVEUHI-Source_DACA_PVEULO > 999)
                                153
                                154
                                                           return "D0";
                                155
                                                       else if(Source_DACA_PVEUHI-Source_DACA_PVEUL0 > 99)
                                156
                                                           return "D1";
                                157
                                                       else if(Source_DACA_PVEUHI-Source_DACA_PVEUL0 > 2)
                                158
                                                           return "D2";
                                159
                                                       return "D3";
                                160
                                                                            Ш
Code View tab
                              Code
```



Click this tab to view any contained source code. If this tab is not visible in the application workspace, then verify that this view option is checked in the Ribbon Toolbar <u>View</u> tab.

#### **Evaluation view**



Source Name

Source Name

The source Property name for the Control Binding.

Source Value

Source Value

The value for the source Property in the Control Binding. The default value specified in the Control Binding [DefaultValueIfEmpty] configuration is used initially. You can change this value to test how target properties evaluate based on their Control Binding expressions.

Target Name

The target Property name for the Control Binding.

**Target Value** 

Target Value

The value for the target Property in the Control Binding. This value is calculated based on the source Property value(s) and the Control Binding expressions.

# **PKS Configuration Views**

Control Architect provides several views for PKS Cxxx based configuration including a raw XML source view, Data View (entity ListView), Chart View, HMIWeb Graphic View, and an SCM# view.

These configuration views provide various analysis tools that facilitate project development, technical assurance activities, and life cycle management of Control strategies.

Control Architect also supports the conversion of other non-PKS CEE type configuration such as Honeywell Exception Build (EB) and SCADA Quick Builder to a compliant PKS Configuration standard. Control Architect uses the PKS CEE Configuration schema as the standard for all configuration views and tools. A Text File conversion tool allows you to convert other non-PKS CEE configuration types to a compliant PKS Schema type.

#### **Xml Source View**

This view provides a raw formatted read-only view of the Xml configuration. The Xml in this view is searchable.

#### **Data View**

This view provides a list of Control Module entities that reside in a Configuration Asset data model that was generated for the Experion Server project. Xml configuration files that have been exported from an Experion Server are parsed to an internal Configuration Asset Data Model using a parse function within this view.

This DataView also supports exporting entity parameter data to Excel using its Data Export Wizard. Entity parameter data scope can be defined along with the column ordering in the exported Excel Workbook file. Excel Worksheet formatting is also supported.

This view supports analysis type functions such as the Peer Graph and Control Graph.

#### Peer Graph

This view provides a peer analysis of all Control Execution type nodes found configured within the Experion Server including Cxxx CEE, ACE CEE, and FIM type nodes. This visual Peer Graph provides a quick view of Peer-to-Peer traffic between nodes that if used during project development will help to guide efforts to reduce the Peer-to-Peer traffic. This Visual Graph also supports LCN UCN and AM Peer.

#### **Control Graph**

This view provides a graphical block overview of Control Modules and their connection relationship dependencies. By default the selected Control Module entity in the data ListView is used to being the connection relationship analysis when the Control Graph view is invoked. This single entity is analyzed for its immediate connection relationships to display a graphical view of a Control Strategy. Each visual block Control Module provides an expand/collapse button that support further visualizing of Control Module relationships by delving into their connection relationships.

#### **HMIWeb View**

This view provides a list of Process Graphic files that have been imported for display of their entity references and report generation. The Process Graphic file references include the embedded shape scripts. Process Graphic file content can be viewed by associating an image with the graphic during import. The included HMIWebCapture tool converts HMIWeb Process graphics to either PDF or PNG file types that can be imported together with the HMIWeb graphic files. Custom reports can be generated to show Shape usage and validation of entity references.

#### **Chart View**

This view provides a Control Builder equivalent Chart View of the Cxxx configuration with many improvements and enhancements including external reference lists (including use in SCM Alias Table), quick ToolTips for viewing external reference list and expressions for expression supported function blocks such as AuxCalc and EnhRegCalc, CAB source code views with syntax color classifications, attached Document references, and a customizable property view. All connection references are navigable along with drop-down lists to navigate to function blocks within the view and also to navigate back and forward to previously viewed modules.

This view also integrates with the SCM# view when viewing an SCM type module. SCM sequence blocks can be navigated to their SCM# code block and SCM# code blocks can be navigated to their Chart View sequence block.

The ChartView also supports an Ink Annotation View that facilitates collaboration with other users.

#### **Ink Annotation View**

The Ink Annotation View provides ink tools that allow users to markup Control Modules within the ChartView to facilitate collaboration with 3rd party integrators, MOC change requirements, etc. The ink annotations can be saved with the ChartView and also saved to an XPS document file.

#### **SCM# View**

Copyright © 2024 by Automation Software Technology, Inc. All Rights Reserved.

Control Architect Help Documentation

This view provides another view of an SCM module translated to SCM# language code. This currently is a read-only view with plans to make this a development capable view using the SCM# language.

# **Stats View**

This view provides statistics for the configuration project including Template usage, Memory Unit and Process Unit calculations.

LCN xPM projects include additional stats related to Control Processor allocation/ loading including I/O Module slot loading.

#### **PKS Views Introduction**

Control Architect releases starting with R4.x deploy a Server for maintaining configuration files, graphic files and their Asset models. The original data model was pure file based, which resulted in having to maintain hundreds of thousands of configuration files in folders on your system. This also resulted in system performance issues when anti-virus applications were required to scan this huge file load.

A new server based document model is now deployed and used in Control Architect. This is a non-relational database typically referred to as a Document database. The database technology that was chosen is a commercial scalable product named MongoDB. During installation a local instance of this Server technology is installed. Users can choose to scale this to a Corporate Server or Cloud installation for an Enterprise model. This product does not require any additional licensing unless you choose to scale to an Enterprise model.

A MongoDB administration tool named <u>Compass</u> is installed to help administer the MongoDB database. This product does not require any additional licensing unless you choose to scale to an Enterprise model.

With the move to this new Document model, it is much easier for users to manage their versions of PKS configuration projects.

- PKS XML configuration and SCADA configuration files are convert to and stored as XML files in the new document server
- HMIWeb graphic files are stored in the new document server.

Control Architect supports a Database/Collection model for storing configuration data. A Database should represent a single PKS Server and will typically be named similar to the Server or Process Unit name such as "VPS" for a Crude unit or "DCU" for a DeCoker unit, etc. A Collection represents a version of PKS XML configuration files that were exported from a PKS Server. A Database may contain multiple Collections or versions of exported configuration files. You will typically name your Collections to include both the Process Unit name and time stamp of the configuration file set. New Database collections that are created are referenced using the format *DatabaseName:CollectionName* where a ':' character separates the database name and collection name components. A typical Database:Collection name would be formatted as in "VPS:VPS\_20170110".

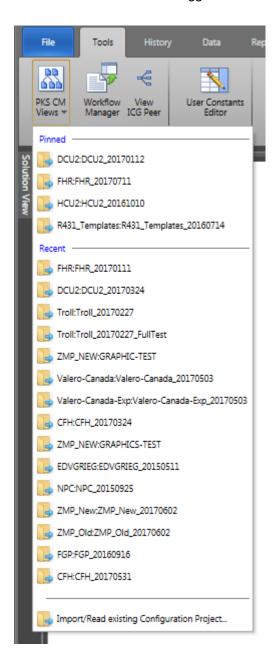
During the import process your PKS configuration source files are moved to the Document database and Asset models are generated and also stored to the same database. Configuration entities and graphic source files can later be exported from within Control Architect to generate their original XML and binary based \*.cnf.xml/\*.bcd files that can be imported back into your PKS Server using Control Builder. HMIWeb Graphic files cannot also be exported to their original file contents.

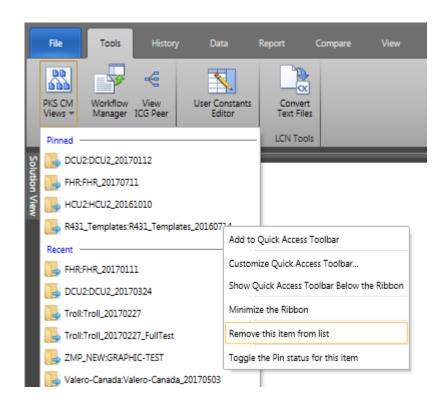
# **Open existing PKS Configuration project**

Configuration projects when created or opened are saved to a Most Recently Used (MRU) list. This list supports pinning of project items to keep the items in the list until unpinned or removed. The number of items saved to this MRU list is a configurable option accessible from the <u>File</u> <u>Toolbar menu</u>.

Each item in this MRU list has options that are accessible through its right mouse-click context menu. Common options for MRU list items are:

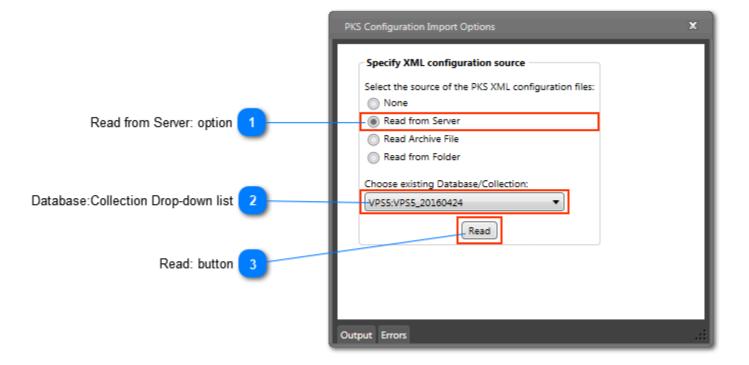
- Remove this item from list removes the project item from the MRU list
- Toggle the Pin status for this item toggle the Pin status of this item and moves it to the Pinned category.





### Reading an existing Project from the Server

If your project is no longer accessible from the MRU list of projects, you can read the project directly from the Server.



Read from Server: option

Read from Server

Choose this option to open an existing PKS configuration project that exists in the Document database.

Database:Collection Drop-down list

VPS5:VPS5\_20160424 ▼

Click this drop-down to view the available list of configuration projects that have been imported to the Document database.

Read: button



Click this button to read or open the selected configuration project.

# Migrating to new document Server

Control Architect now supports new document Server technology named MongoDB for maintaining configuration files. This Server technology is installed when you install Control Architect. This new document Server will maintain the thousands of configuration files that in the past had to be maintained and organized in your local file system. The topics below will provide information that will help guide you in migrating to the new Server technology.

## **Existing MRU Project references**

The existing MRU Project reference menu may still show references to their folder paths in your local file system as indicated below:



Project references that include a folder separation character '\' are no longer accessible and will fail to open. In the example above the project reference F:\FHR\Boilder\BLR\_20170111 is invalid. The new Project reference format is also visible above where the reference is formatted using the new ':' notation as in HCU2:HCU2\_20161010 where the item to the left of the ':' separator char is the Database name (typically the PKS Server or process Unit name) and the item to the right of the ':' separator char is the Collection name (typically a version export from PKS Server).

Right mouse-click on any invalid project reference and choose the menu item **Remove this item from list** to remove the invalid project reference.

As you migrate your existing projects to the Server you should manually remove the original folder path type reference.

#### **Existing PKS Configuration files**

When migrating to the new document Server technology it is no longer necessary to save uncompressed PKS Server exported XML configuration files in your local file system. During import to the Server all required configuration type files are copied to the specified Server Database:Collection. These configuration files include \*.cnf.xml, \*.bcd, and other file types associated with CAB and Template types. It is recommended that you maintain an archive file (compressed) that was exported from the PKS Server. The Server import function can read archive file types for importing of configuration type data.

#### **Existing HMIWeb Process Graphic files**

Unlike PKS configuration files, HMIWeb graphic files at this time <u>are not</u> stored to the document Server. You are still required to maintain these graphic files on your local file system. During import of your HMIWeb graphic files, the graphic files are analyzed and a graphic Asset model is generated and stored to the document Server. No graphic files are currently stored to the Server. This inconsistency exists because of the need to read the graphic data and shape type content files that typically make up a process graphic when viewing the graphic in a Browser view.

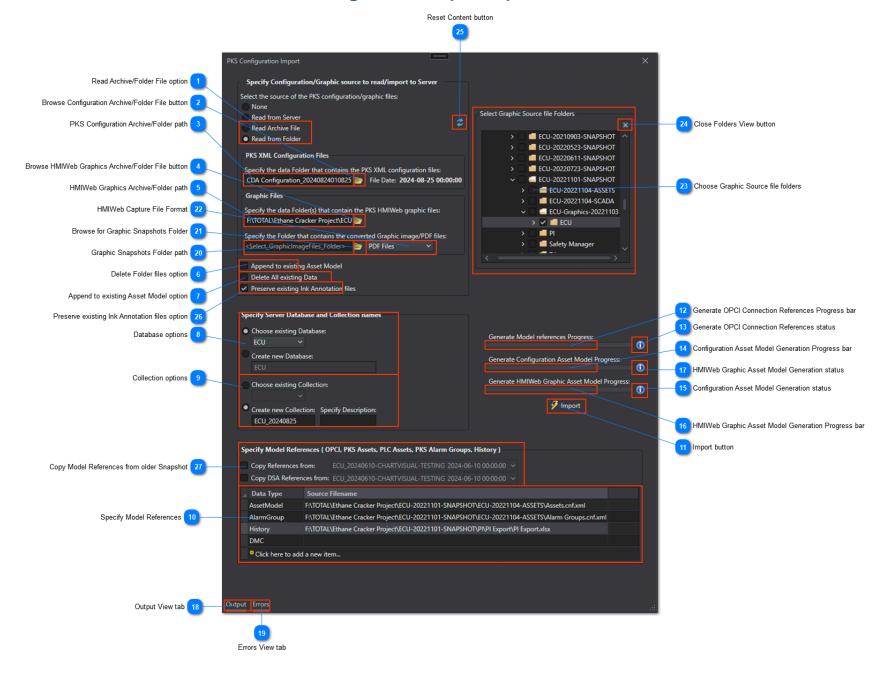


The obsolete project reference indicated the folder path where your configuration files were stored.

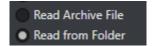


The new project reference indicates the DatabaseName:CollectionName where your configuration files are stored in the Server.

# **PKS Configuration Import Options window**



### Read Archive/Folder File option



Choose the *Read Archive File* option to import configuration data that is currently stored in an archive file type such a \*.rar or \*.zip or choose *Read from Folder* option to import data that currently exists in a file folder.

- During the import process, the files contained in the archive file are un-compressed and stored to a temporary folder that is later deleted after completion of the import operation.
- When importing from an existing folder, you can use the option *Delete Folder files* to delete the folder and its file contents when the import operation has completed.

# Browse Configuration Archive/Folder File button



Click this button to browse for an existing Folder path or Archive file that contains your configuration source files that were exported from a PKS Server.

# PKS Configuration Archive/Folder path

CDA Configuration\_20240824010825

Displays the selected Archive file or Folder path.

#### Browse HMIWeb Graphics Archive/Folder File button



Click this button to browse for an existing Folder path or Archive file that contains your HMIWeb graphic files that were copied from your PKS Server abstract folder. The graphic \*.htm files and their same named containing folders are required to successfully parse and generate an Asset model.

#### HMIWeb Graphics Archive/Folder path

F:\TOTAL\Ethane Cracker Project\ECU

Displays the selected Archive file or Folder path.

# Delete Folder files option

Append to existing

Choose this option to delete the folder path and its contained files after importing to the Document database. Using this option will help minimize the amount of configuration files that are stored in your local file system.

# Append to existing Asset Model option

Delete All existing Data

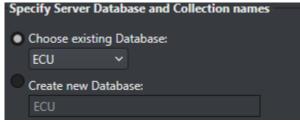
Choose this option to append XML configuration data into the existing Asset Model. If this option is chosen, then any existing configuration and graphic data is retained and the imported configuration is added. If this option is not chosen, then the existing referenced Database collection and its Asset model is deleted prior to importing the specified XML configuration.

You cannot append to existing HMIWeb graphic data as the retained graphic Asset Model will be generated based on the source files that are parsed during this task.



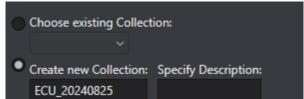
You will typically use this option to support the importing of HMIWeb graphic or additional PKS XML configuration to an existing Database Collection (Project) that was created during an earlier import operation. If you are adding PKS Server Scada configuration to an existing Database Collection, the Text File conversion tool supports the same "Append..." option that must be chosen to add the new data to the existing project.

# Database options



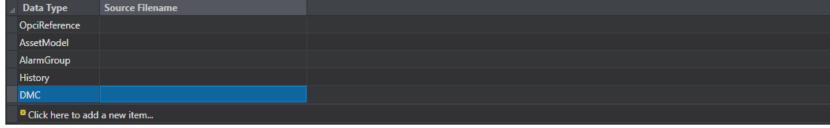
Choose either an existing Database name or create a new Database. Typically the name chosen for the database will represent the Process Unit that the PKS Server provides Control functions for. For example a user may choose "VPS" for a typical Crude Unit or "ETH" for a typical Ethylene Unit.

# Collection options



Choose either an existing Collection name or create a new Collection. Typically the name chosen for the collection will include the Process Unit and a time stamp to identify the version of configuration files that are being imported. A default Collection name is generated based on the time stamp found in the source files and prefixed with the specified Database name as in "VPS5\_20160424". When defining this Collection name take into consideration that you may be storing multiple versions (collections) of PKS configuration to this same database.

# Specify Model References



This import process supports OPCI configuration, PKS Asset Model, PKS Alarm Groups, Both PI and PHD history, and DMC that have been exported from the respective application hosts.

When specifying OPCI data files you will have to retrieve the OPCI configuration files below from your PKS Server. OPC Integrator data files stored in <a href="mailto:stored-no-en-alpha-16">stored in <a href="mailto:stored-no-en-alph

Add the following file references:

- 1. OPCI\_Online\_Groups.tsv
- 2. OPCI\_Online\_Items.tsv



- You must provide both the OPCI Groups and Items file as references.
- PKS Asset Model and Alarm Groups are exported from the EMDB Server.
- PI/PHD History and DMC configuration data files are exported from their respective application hosts.

# Import button



Click this button to begin the Import process. After an import has been executed you can again click this button to cancel the operation.

# Generate OPCI Connection References Progress bar

The current progress for generating the referenced PKS OPCI configuration source files for generation of the Configuration Asset model.

# Generate OPCI Connection References status



Displays the current status of the OPCI connection generation task.

# Configuration Asset Model Generation Progress bar

The current progress for importing the referenced PKS configuration source files to the Document database and generation of the Configuration Asset model.

# Configuration Asset Model Generation status



Displays the current status of the Configuration Asset model generation task.

# HMIWeb Graphic Asset Model Generation Progress bar

The current progress for generating the HMIWeb graphic Asset model.

# HMIWeb Graphic Asset Model Generation status



Displays the current status of the HMIWeb graphic Asset model generation task.

# Output View tab



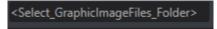
Click this tab to view information associated with the Import process.

#### Errors View tab



Click this tab to view any errors or warnings encountered during the Import process.

#### 🥋 Graphic Snapshots Folder path



Displays the selected snapshots Folder path.

#### 👧 Browse for Graphic Snapshots Folder



Click this button to browse for an existing Folder path that contains your HMIWeb graphic snapshot files that were generated using the **HMIWebCapture** tool. The snapshot files that are generated from the HMIWeb Display Builder static rendering will be either PNG Image or PDF Acrobat files. The files that are referenced here will be associated with a Graphic item that can then be viewed from the Configuration views HMIWeb View tab. Choose the file format to search for.



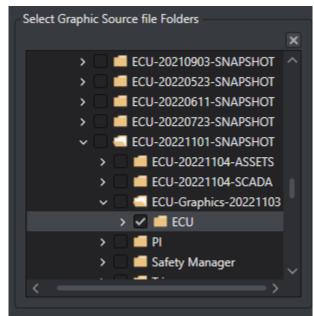
## **HMIWeb Capture File Format**



Choose the file format to search for in the specified Graphic Snapshot folder. The HMIWebCapture tool allows you to choose to capture the HMIWeb Display Builder view as either PNG/BMP Image or PDF Acrobat files. The format chosen here will be imported to the Server and automatically associated with its HMIWeb Graphic file using the name of the snapshot file.

23

## **Choose Graphic Source file folders**



Check the folders that contain the HMIWeb graphic \*.htm source files that are to be imported.

This view becomes visible when you click the graphic files browse button.



## **Close Folders View button**



Click this button to close the Folders View. You can reopen it by again clicking the graphic files browse button.

25

#### **Reset Content button**



Click this button to clear the existing content. Use this button when you need to execute another import task using a different set of source files.

26

#### Preserve existing Ink Annotation files option

Preserve existing Ink Annotation

Check this option to preserve existing Ink Annotations and saved Control Graph View states when overwriting existing Database:Collection data.

27

#### Copy Model References from older Snapshot



Choose the option to copy Model references and State data including Control Graph and Ink Annotation states from another older Database Collection configuration snapshot. The History Model Reference cannot be copied from another project. It must be specified as an additional source reference for importing. It is recommended that new Collection snapshots be created with the latest available data from your Online System. This option should only be considered as a convenience when it is known that some Model reference data hasn't changed such as PKS Assets or PKS Alarm Groups. You can choose this option, but still specify a Model Reference in the Data Grid to take precedence over the copied reference.

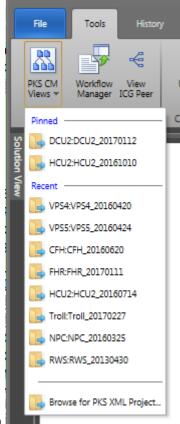
Choose the option to copy DSA references from an older Database Collection configuration snapshot.

# **Getting Started Using PKS Views**

Control Architect features work with a Configuration Asset Model that is created from XML configuration files that have been exported from an ERDB, LCN, or Quick Builder source. Follow the steps below to generate a Configuration Asset Model and begin working with the Control Architect configuration views and PolicyCop.

# Using Control Architect for a PKS Cxxx/ACE Project

- 1. Export XML configuration files from your EPKS Server ERDB.
  - i. Using Control Builder, Upload then Update the project side configuration prior to exporting.
- 2. It is recommended to name the folder containing a particular set of Server configuration files in a form that identifies the date of the export as in *VPS\_20140325* where the Process Unit identifier is followed by the year 2014, followed by the month 03, and followed by the day 25.
- 3. Launch Control Architect and click the **Tools** Toolbar button **PKS CM Views.**



- 4. Select the **Browse for PKS XML Project...** menu item.
- 5. Use the Import options to import your PKS configuration data to the Server. This will create a new Configuration project that will be formatted as in DatabaseName:CollectionName.
- 6. Using the same drop-down menu noted in item 4 above, select the created project from the MRU list. The <u>Data View</u> tab will be displayed by default that contains the entities that were imported from the configuration files.

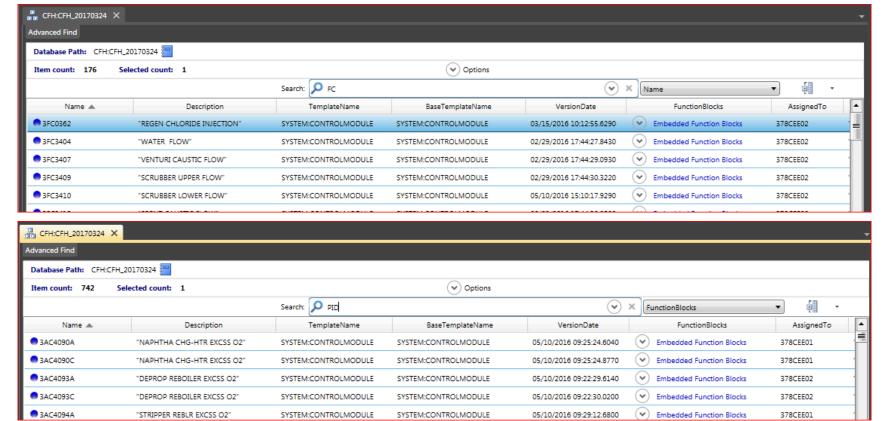
### Using Control Architect for a PKS SCADA or LCN Project

- 1. Export SCADA or LCN Exception Build (EB) configuration files from Quick Builder or LCN Native Window.
  - a. In Quick Builder choose the menu item *File-Export*... to export your SCADA configuration using the "*Hardware/Point build format*". Select the Scope option "*All Items relevant to file type*" to export all items including Channels, Controllers, and points to \*pnt & \*.hdw file types.
- 2. It is recommended to name the folder containing a particular set of configuration files in a form that identifies the date of the export as in *VPS\_20140325* where the Process Unit identifier is followed by the year 2014, followed by the month 03, and followed by the day 25.
- 3. Launch Control Architect and click the **Tools** Toolbar button **Convert Text Files** to display the **Conversion tool**.
- 4. Select the Text files to convert to a XML configuration. You can choose either file extension types \*.eb or \*.pnt/\*.hdw.
- 5. Specify the root folder where your LCN Control Language (CL) files reside. If the files reside in multiple subfolders then check the option "Use SubFolders".
  - a. CL files specified in the folders are analyzed for connection relationships during the conversion process.
- 6. Browse to select or create the database:collection where your converted to XML configuration files will be imported to.
- 7. Select the Control System Target Platform option. The option chosen here is used to validate the generated XML configuration file Schema against the chosen platform.
- 8. Leave the option "Overwrite existing XML Configuration files" unchecked. This option would be used if you had to regenerate the same configuration files to the same database:collection.
- 9. Leave the option "Remove existing XML Configuration files" unchecked. This option would be used to clean a database:collection of XML configuration files that were generated from a prior conversion activity.
- 10. Control Architect is installed with a default set of Template styles and attributes for LCN and SCADA Point types. You can view the current style set by clicking the "Specify Template Styles and Attributes" expander.
- 11. Generate the *Configuration XML Files* by clicking the **Convert** button.
- 12. When the conversion task is complete, close the tool view window and click the <u>Tools</u> Toolbar button **PKS CM Views** to open an existing configuration project or select a new configuration project.
- 13. Click the <u>Stats View</u> tab to calculate statistics for the configuration project. When this view tab is first selected, it will automatically begin executing the statistics analysis.
  - a. LCN projects that contain xPM type Process Control nodes support visual display of Control Processor Point types

b. LCN projects support copying both the visual Control Processor Point types and their entity slot allocation to the Windows Clipboard for pasting into Excel.

# Displaying a ChartView for a PKS Control Module

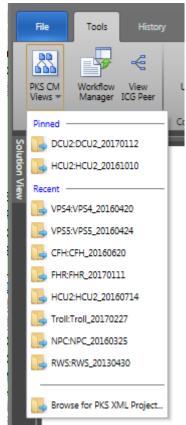
- 1. To display a *Chart View* for a Control Module item located in the <u>Data View</u> tab, double click an item in the *ListView* with your left mouse button.
- 2. Chart View Function blocks will display an *Adorner* when the mouse cursor hovers over them. This *Adorner* contains clickable buttons that display additional information related to the Function Block depending on the type of block.
  - a. All UI Visual elements will display a ToolTip that indicates the use of the element.
  - b. All Function blocks contain parameters that can be viewed by clicking the button located in the top left corner of the block.
  - c. You can resize the Parameters View and move it to another location on the canvas.
  - d. You can open multiple Parameter views.
  - e. You can view expressions for Calculation type blocks such as AuxCalc as a ToolTip by hovering your mouse cursor over the button target or click the button to view the expressions in a View window.
  - f. You can view External references as a ToolTip by hovering your mouse cursor over the button target or click the button to view the references in a View window.
  - g. You can view CAB VB Source code by clicking the button located in the bottom left of the Adorner. This will display a VB Source Code view.
- 3. You can quickly view External references for Control Module items located in the Data View list by hovering your mouse cursor over the blue colored ellipse image located to the left of the Control Module Name column.
- 4. You can quickly filter Control Module items in the ListView by entering text into the Filter box. By default the ListView *Name* column is chosen as the column to be filtered. Entering text such as "FC" will quickly filter the items such that only Control Modules whose Name contains the text "FC" will be displayed in the list. You can also quickly filter on use of Function Block names by choosing *Function Blocks* from the Filter column drop-down list and entering text such as "PID" in the Filter box. This would filter the ListView such that only Control Modules that contain *Function Blocks* whose name contains the specified text will be displayed.



# **Opening a new PKS Configuration Project**

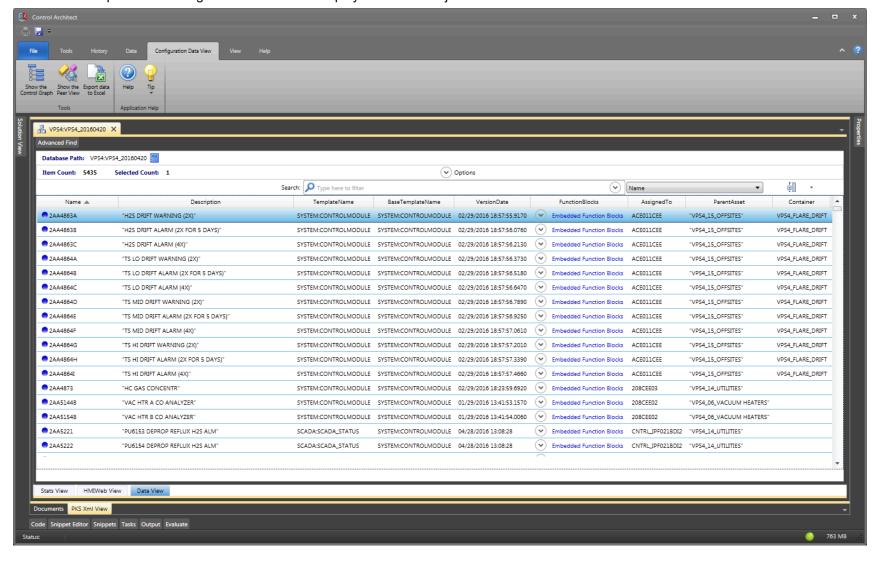
To open a PKS configuration project:

1. Click the PKS CM Views toolbar button in the Ribbon Toolbar.



- 2. Click the menu item Browse for PKS Xml Project... in the drop-down list.
- 3. An Import view will be displayed that allows you to create a new Document database and generate the configuration Asset models.
- 4. When the Configuration Asset Model generation has completed, you can close out the Import view and select the created configuration Database:Collection configuration project from the MRU list of projects.

This is an example of the Configuration view that is displayed when a Project item is selected.



# **Managing Configuration Projects MRU List**

Control Architect supports the standard Ribbon Quick Access Toolbar (QAT) where you can choose to add commonly used items from the application toolbar.

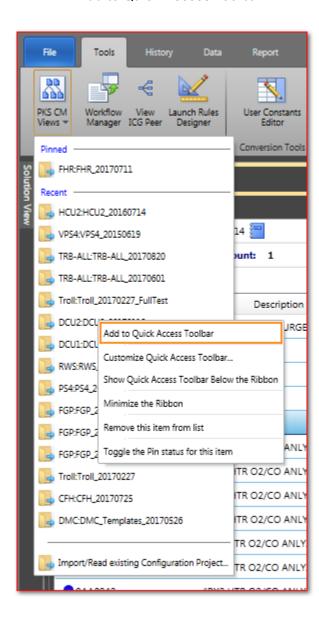
Configuration project Database:Collection paths are stored to a Most Recently Used (MRU) list that is saved and restored when the application is opened. You can define how many items to retain in this list in the <u>Application Options</u> view. The most recently used project path is saved as the first item in the list and the other items are rolled to the end.

The Quick Access Toolbar (QAT) is located above or below the Application Toolbar, which is user defined.



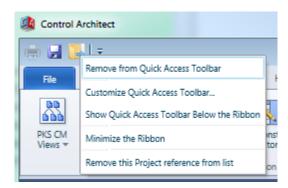
#### Add project paths to Quick Access Toolbar (QAT)

To add a project path to the Quick Access Toolbar, right mouse-click on a project path item in the list to display its context menu. Select the menu item **Add to Quick Access Toolbar**.



#### Remove project path from Quick Access Toolbar (QAT)

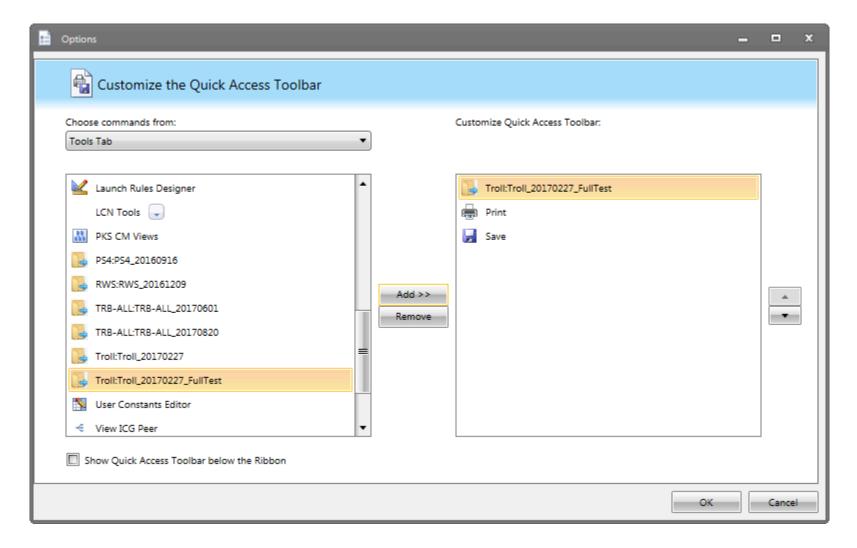
To remove an item from the QAT, right mouse-click on the QAT item to display its context menu. Select the menu item **Remove from Quick Access Toolbar**.



#### **Customize items in the Quick Access Toolbar (QAT)**

You can customize the Quick Access Toolbar by right mouse-clicking on a Toolbar item or any list item from the Toolbar to display its context menu. Select the menu item **Customize Quick Access Toolbar**.... This will display a Customize view that allows you to add/remove and order items in the QAT.

Copyright © 2024 by Automation Software Technology, Inc. All Rights Reserved.

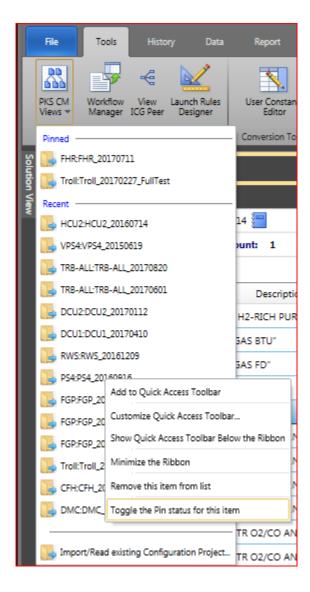


#### Remove a project path from the Most Recently Used (MRU) list

You can remove a project path from the MRU list by right mouse-clicking on a project path item to display its context menu. Select the menu item **Remove this Project reference from list**. This only removes the project path from the MRU list and does not delete any content from the file system.

#### Pin a Project to the MRU List

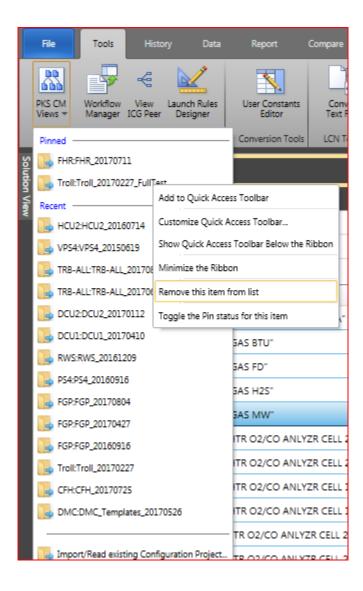
You can pin a project to the MRU list, which prevents it from being removed when the MRU list count is exceeded. Right mouse-click on a project item to display its Context menu and select the *Toggle the Pin status for this item* menu item.



#### **Unpin a Project from the MRU List**

Copyright © 2024 by Automation Software Technology, Inc. All Rights Reserved.

You can unpin a previously pinned project from the MRU list. Right mouse-click on a project item in the Pinned menu group to display its Context menu and select the *Remove this item from list* menu item.

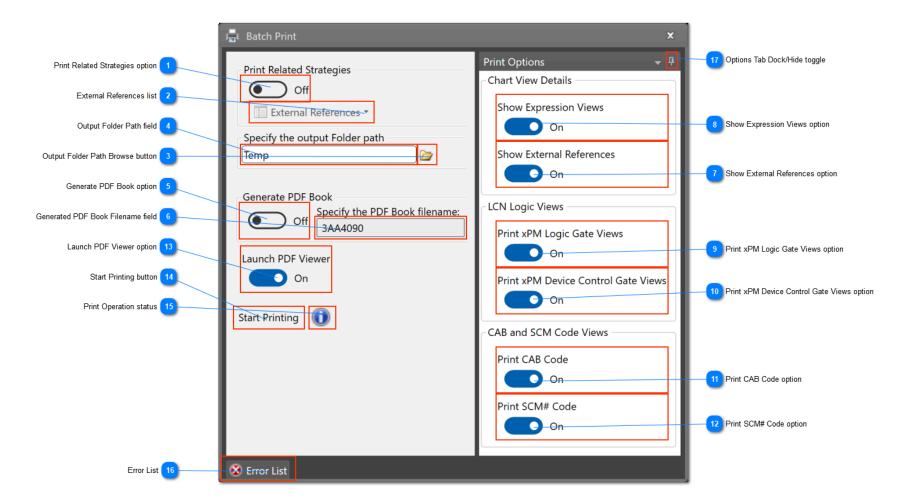


# **Batch Printing**

Control Architect supports the batch printing of configuration entities selected in the configuration Data View. Documents are printed to a PDF file type. The document types that are supported for printing include:

- Chart View
- SCM# Source
- CAB Source
- LCN CL Source
- LCN xPM Logic and Device Control gate views

Users can choose to print to individual PDF files or to a PDF Book file where embedded documents are navigable through Book Marks generated in the PDF Book document file.



# Print Related Strategies option



Choose this option to quickly generate a list of external references to the current scope of selected CM entities to print. These external references are then automatically added to the scope of CM entities to print.

# 2 External References list External References \*

Click this dron-down list of external references to view

Click this drop-down list of external references to view the additional scope of related strategies that will be printed when using the *Print Related Strategies* option.

Users can choose to enable/disable entity items in this list for printing.

# Output Folder Path Browse button



Click this browse button to specify the output folder path where all PDF file content will be generated to.

# Output Folder Path field Temp

Click the browse button to specify the output folder path where all PDF file content will be generated to.

# Generate PDF Book option



Choose this option to print all documents to a single PDF Book file where embedded document items are navigable using generated Book Marks

# Generated PDF Book Filename field

3AA4090

Specify the name of the PDF Book file that will be generated when using the Generate PDF Book option.

# Show External References option

**Show External References** 



Choose this option to expand the External Reference tags to view their full connection reference prior to printing the Chart View content.

# Show Expression Views option

**Show Expression Views** 



Choose this option to show all expression type views typical of AuxCalc block types prior to printing the Chart View content.

# Print xPM Logic Gate Views option

Print xPM Logic Gate Views



Choose this option to print Logic Gate views for all LCN xPM Logic type CM entities.

# Print xPM Device Control Gate Views option

Print xPM Device Control Gate Views



Choose this option to print Logic Gate views for all LCN xPM Device Control type CM entities.

#### Print CAB Code option

Print CAB Code



Choose this option to show all CAB source views for all CAB type blocks embedded in the CM prior to printing the Chart View content.

#### Print SCM# Code option

Print SCM# Code



Choose this option to print the SCM# source code view for SCM/RCM entity types.

#### Launch PDF Viewer option

Launch PDF Viewer
On

Choose this option to send the generated PDF file content to your registered PDF content viewer.

#### 🥋 Start Printing button

**Start Printing** 

Click this button to start printing the selected items in the Data View. While printing is active you can cancel the Print operation by again clicking this same button.

# Print Operation status



Displays the Print operation status.

# 16 Error List



Hover over this Tab with your Mouse cursor or left mouse-click to view errors associated with the current Print operation.

If any errors are encountered while printing, they will be displayed in this list while this Tab will flash red indicating an error exists.



# **Options Tab Dock/Hide toggle**



Click this pin image to toggle the Dock status of the Options Tab view.

#### **Chart View**

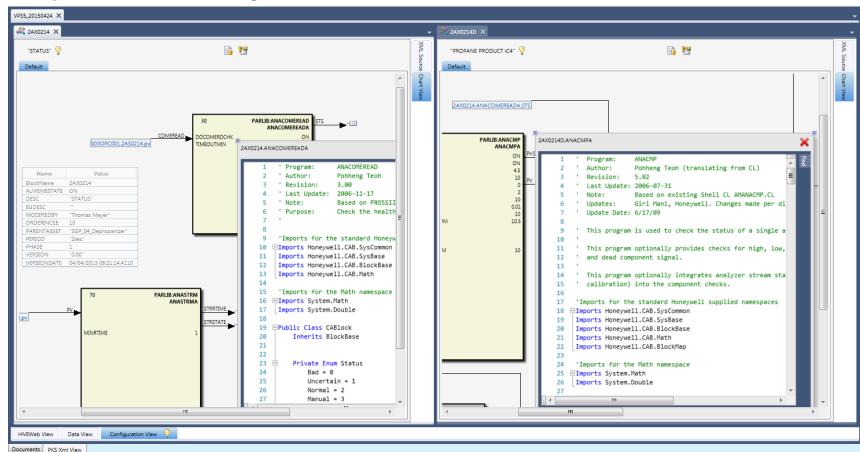
This view provides a Control Builder equivalent Chart View of the Cxxx configuration with many improvements and enhancements including external reference lists (including use in SCM Alias Table), quick ToolTips for viewing external reference list and expressions for expression supported function blocks such as AuxCalc and EnhRegCalc, CAB source code views with syntax color classifications, attached Document references, and a customizable property view. All connection references are navigable along with drop-down lists to navigate to function blocks within the view and also to navigate back and forward to previously viewed modules.

This view also integrates with the SCM# view when viewing an SCM type module. SCM sequence blocks can be navigated to their SCM# code block and SCM# code blocks can be navigated to their Chart View sequence block.

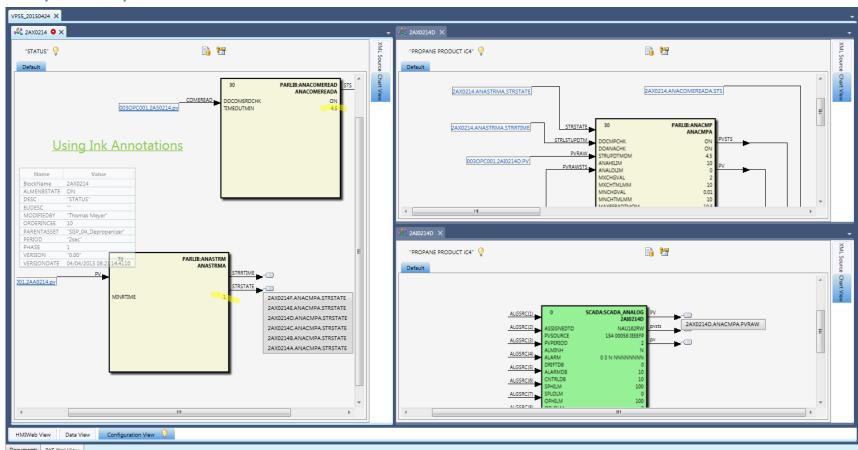
The ChartView also supports an Ink Annotation View that facilitates collaboration with other users.

Multiple views were introduced in release R2.20, which supports unlimited Chart views and advanced layout & docking,

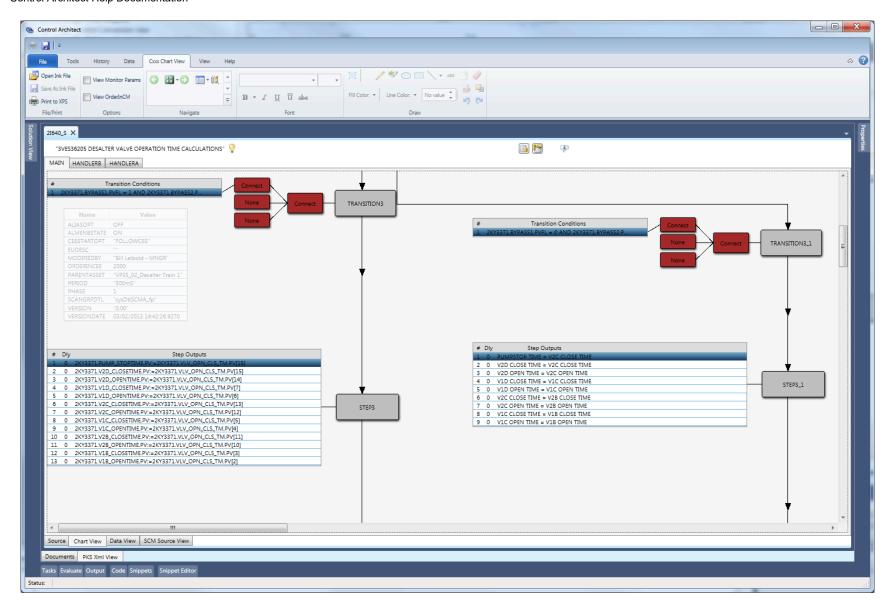
#### **Example of a Control Module using CAB.**



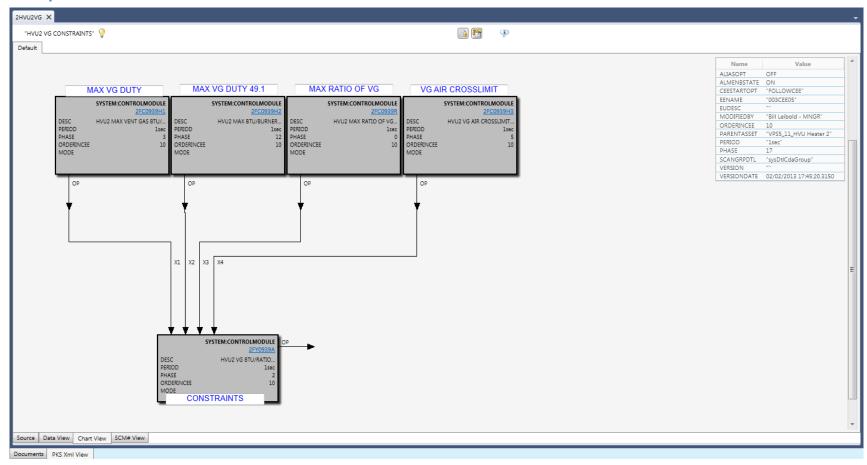
#### **Example of multiple docked views and Ink Annotation**



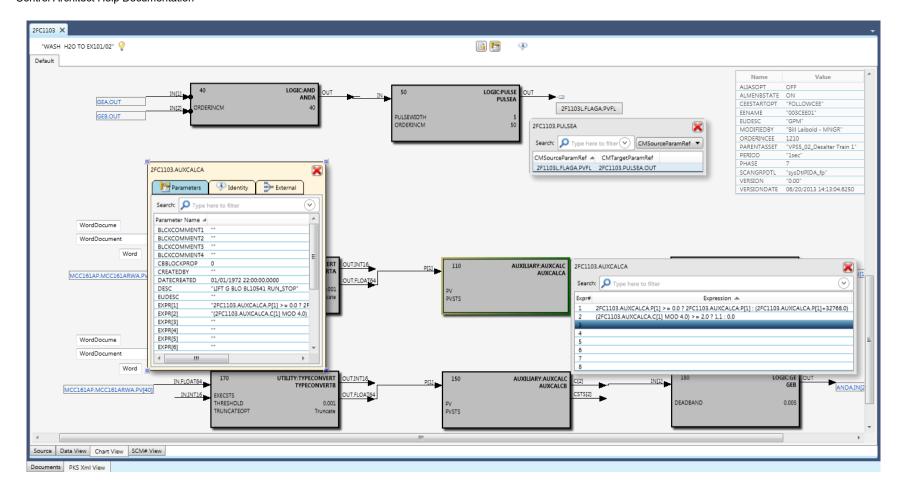
# **Example of an SCM Module.**



#### **Example of Hierarchical Embedded Control Module.**



Example of Control Module with various Function Block views and the options [View Monitor Parameters] and [View OrderInCM].



#### **Tips & Tricks**

- Left mouse double-click in the ChartView canvas (not on a function block visual) will automatically display the Control Module properties view.
- Pressing the key combination Ctrl+R will toggle the visibility for all function block external reference views.
- Pressing the key combination Ctrl+E will toggle the visibility for all expression view for function block types that support expressions including AuxCalc, EnhAuxCalc, RegCalc.
- Pressing the key combination Ctrl+T will toggle the visibility for all external reference type tag shapes.
- Pressing the key combination Ctrl+C for items displayed in list type views will copy the selected items to the Windows clipboard for pasting into Excel.

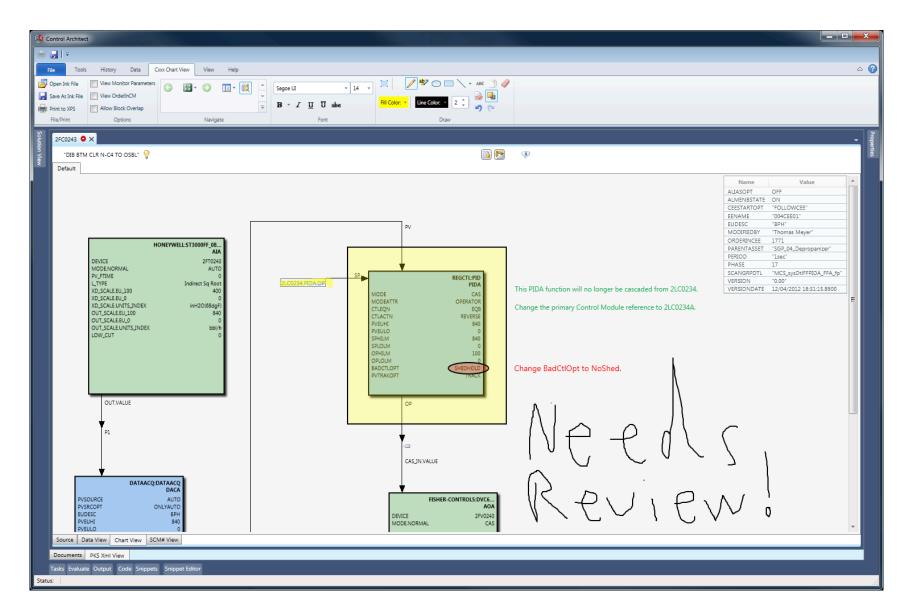
#### **Annotation View**

The Ink Annotation View provides ink tools that allow users to markup Control Modules within the ChartView to facilitate collaboration with 3rd party integrators, MOC change requirements, etc. The ink annotations can be saved with the ChartView and also saved to an XPS document file.

Ink Annotation mode is invoked in a ChartView by clicking the **Annotate** Toolbar button.

Configuration items that contain Ink are noted by a status in the Data View. A mouse hover over the status image indicates that the CM item contains Ink annotations.





# **Working with Ink**

#### **Common Ink Functions**

All Ink shapes that can be added to the Ink canvas can be edited to change their attributes such as size, colors, font, line style, etc. Selected Shapes can also be moved to change their position on the Ink canvas.

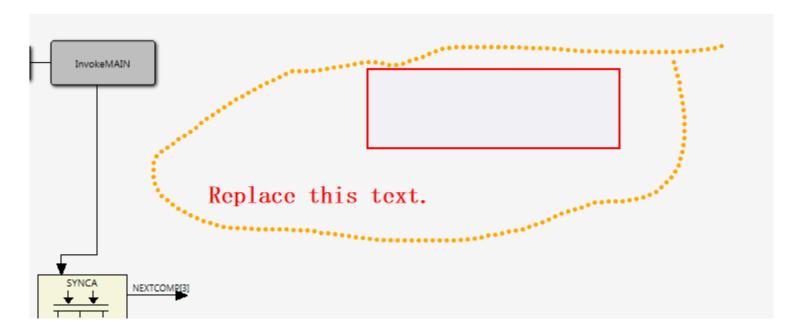
#### Selecting Ink shapes

To edit an existing shape or group of shapes, you must first select the shape or shapes to be edited.

You must first click the Selection Toolbar button to display the selection cross cursor.



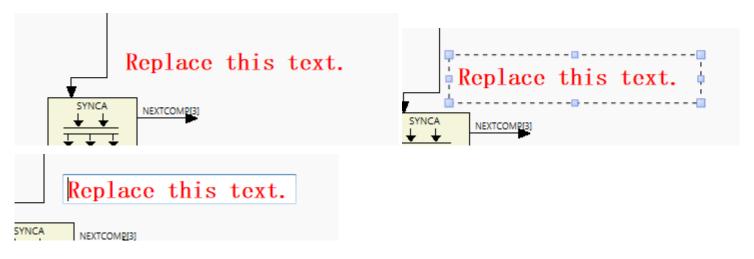
With the selection cursor active you can choose to try and select a shape by clicking in its center with the left mouse key or dragging the selection cursor on the canvas to draw a selection boundary. Left mouse-click on the lnk canvas and while holding the left mouse key down, drag a selection boundary around the shape or groups of shapes to be selected. If a selection boundary intersects a shape, then that shape will not be selected.



#### **Edit existing Text shape**

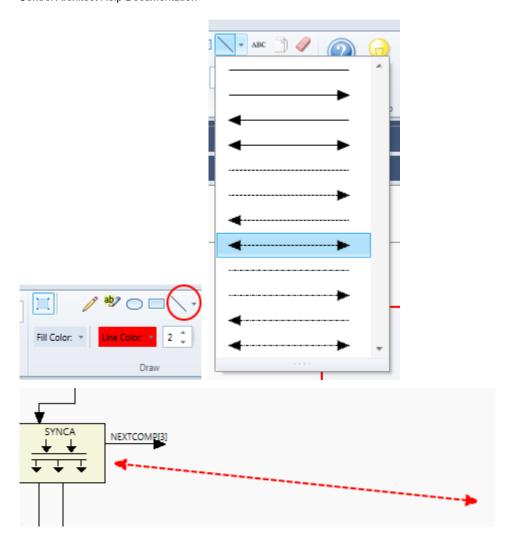
When editing an existing lnk Text shape, you must first select it using the lnk selection cursor. Press the key combination Ctrl+Enter on the selected Text shape to change the lnk Text shape to a Text Box Control element that can then be edited to change its text value or other attributes such as font, text stroke color, size, etc.

This sequence of images shows the unselected Text shape, the selected Text shape with its selection rectangle, and the Text shape in its edit mode when pressing Ctrl+Enter.



#### **Drawing Line shapes**

To draw a line shape with arrow ends select the type of line style to use from the Toolbar. Left mouse-click on the lnk canvas and while holding the left mouse key, drag the line shape to its desired size.



#### **Erasing Ink shapes**

You can quickly erase existing Ink from the canvas by selecting the Ink Eraser from the Toolbar and swiping shapes using a left mouse-click drag motion on the canvas shape. You can also undo/redo strokes by pressing the key combinations Ctrl+Z and Ctrl+Y. You can also press the Delete key to remove selected Ink shapes from the canvas.



# **Ink Annotation options**

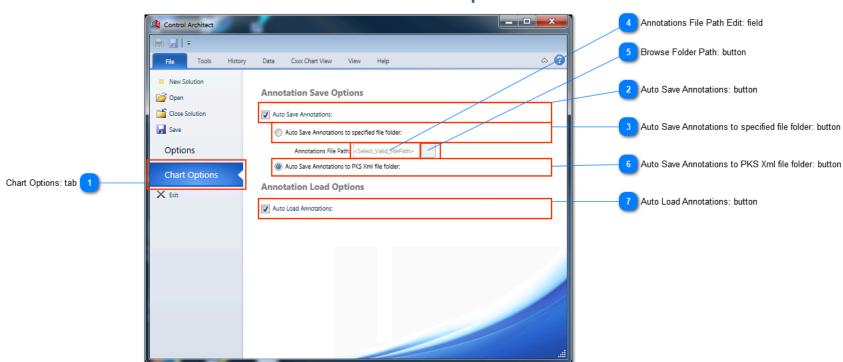
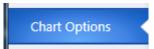


Chart Options: tab



Select this tab to configure Chart View Ink Annotation options.

Auto Save Annotations: button

✓ Auto Save Annotations:

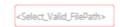
Choose this option to auto-save any Ink annotations made to rendered Chart views. When the annotated ChartView is closed or navigated away from, the Ink annotations will automatically be saved to an Ink file that resides in the same XML folder path where the rendered XML Configuration file resides or another selected folder path. If the *Auto Load* option is chosen, then the next time this Chart View is rendered it will automatically load its Ink annotations.

Auto Save Annotations to specified file folder: button

Auto Save Annotations to specified file folder:

Choose this option to save the Chart View Ink annotation files to the specified folder path. Click the Browse button to specify the folder path.

Annotations File Path Edit: field



This is a read-only Edit box that displays the selected folder path. Hover your mouse over this field to see the full folder path as a ToolTip.

Browse Folder Path: button

Click this button to browse for the folder path where your annotated Ink files will reside.

Auto Save Annotations to PKS Xml file folder: button

Auto Save Annotations to PKS Xml file folder:

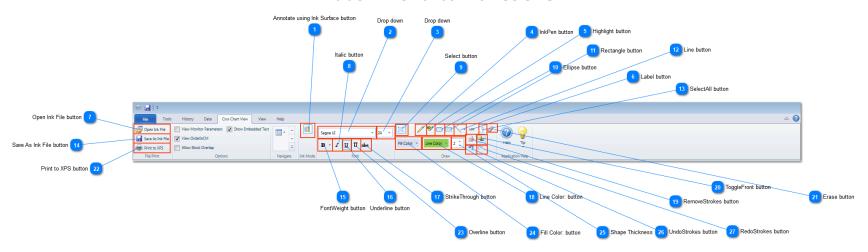
Choose this option to save the Chart View Ink annotation files to the same folder path that contains the XML Configuration files.

Auto Load Annotations: button

✓ Auto Load Annotations:

Choose this option to auto load any existing lnk annotation files associated with a rendered Chart View.

#### Ribbon menu bar functions

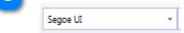


# Annotate using Ink Surface button



Toggle this button to active/deactivate the lnk Surface.

Drop down



Choose the font family to use for text Label type shapes.

Drop down



Choose the font size to use for text Label type shapes.

InkPen button



Click this button to select the lnk Pen as the drawing tool. This allows you to draw lnk using your mouse cursor as the pen.

Highlight button



Click this button to select the Highlighter as the drawing tool. This allows you to draw highlights on the ChartView image. The size of the highlight stroke can be changed by adjusting the shape thickness size.

Label button



Click this button to select the Label as the drawing tool. This allows you to drop a text object onto the lnk surface. When a Label is added to the lnk surface a default text phrase is added that you will replace with your own text value. To complete the edit session click outside the label shape in the lnk surface background. You can edit and change the text value or font attributes for a Label by first selecting the Label shape and then pressing the key sequence *Ctrl+Enter*.

Open Ink File button



You can open a previously saved lnk file.

Italic button



Toggle this button to use Italic font for the Label shape being edited.

Select button



Click this button to use the Select tool. This tool allows you to select one or more lnk surface shapes. Shapes can be selected by finding the origin point for the shape using a left mouse-click or by using the mouse cursor to draw a closed polygon around a shape or group of shapes.

Selected shapes can have their properties such as size, position and colors changed together as a group.





Click this button to use the Ellipse tool.

# Rectangle button



Click this button to use the Rectangle tool.

# Line button



Click this button to use the Line tool. This button will display a drop-down list of various line shapes that can be chosen for the drawing tool.

# SelectAll button



Click this button to select all shapes on the lnk surface.

#### Save As Ink File button



Click this button to save the Ink surface to an external Ink file.

# FontWeight button



Click this button to select the font weight for Label type shapes.

#### Underline button



Toggle this button to select an Underline font.

# StrikeThrough button



Toggle this button to select a StrikeThrough font.

#### 📆 Line Color: button



Click this button to select the color to use for lines, shape borders, highlighting, ink and labels.

#### RemoveStrokes button



Click this button to remove all selected lnk strokes.

# ToggleFront button



Click this button to toggle the ChartView surface in front of the Ink surface.

#### Erase button



Click this button to use the Eraser tool.

# Print to XPS button



Copyright © 2024 by Automation Software Technology, Inc. All Rights Reserved.

Click this button to print the Ink surface to an XPS Document file.

Overline button

Ū

Toggle this button to select an Overline font.

Fill Color: button

Fill Color: \*

Click this button to select the color to use for Ellipse and Rectangle shape fills.

Shape Thickness

2 \*

Select the size of stroke to use when drawing shapes.

UndoStrokes button

7

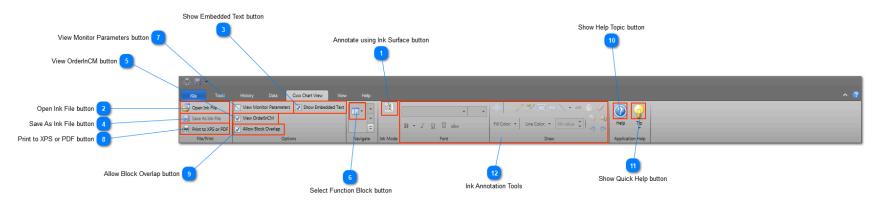
Click this button to undo the last stroke(s).

RedoStrokes button

G

Click this button to redo the last stroke(s).

#### ChartView Overview



# Annotate using Ink Surface button



Click this button to enable/disable the Ink Annotation surface. When enabled, you can annotate the Chart view using the Ink annotation functions.

# Open Ink File button



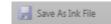
Click this button to open a \*.ink file that contains Ink strokes previously used to annotate the Chart view.

### Show Embedded Text button



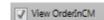
Choose this option to view any embedded Text type objects in the Chart view. Embedded Text type objects include Microsoft Word and WordPad instances.

# Save As Ink File button



Click this button to save the annotation lnk strokes to an external \*.lnk file.

#### View OrderInCM button



Choose this option to view the OrderInCM parameter value on the Function Block surface.

#### Select Function Block button



Click this drop-down button to view a list of Function blocks embedded in the Control Module. Click an item in this list to navigate to its block shape in the ChartView.

# View Monitor Parameters button



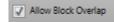
Choose this option to view the Monitor type parameter set on the Function Block surface.

#### Print to XPS or PDF button



Click this button to print the contents of the Chart view to an XPS or PDF document file.

#### Allow Block Overlap button



Choose this option to allow small block overlaps when the Chart view is visually constructed from the PKS XML configuration file. At times blocks may overlap due to the way the ChartView was saved in Control Builder. If this option is not chosen and overlapping is detected, then the Chart view drawing function will attempt to reposition the block shape so that it does not overlap. This option is useful when viewing I/O type modules where in Control Builder they don't have a visual representation.

#### Show Help Topic button



Click this button to view the Help document topic for Chart View.



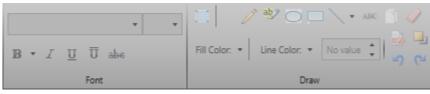
# **Show Quick Help button**



Click the drop-down arrow to view quick <u>help information</u> for Chart View.

12

#### **Ink Annotation Tools**



Tools to support Ink Annotation on Chart View surfaces.

#### **ChartView Help Tips**



#### **General Help Tips**

Ctrl+R Use this command to toggle the visibility for all function block external reference views.

Ctrl+E Use this command to toggle the visibility for all function block expression views.

Ctrl+C Use this command within any list type views to copy the selected items to the clipboard for pasting into Excel.

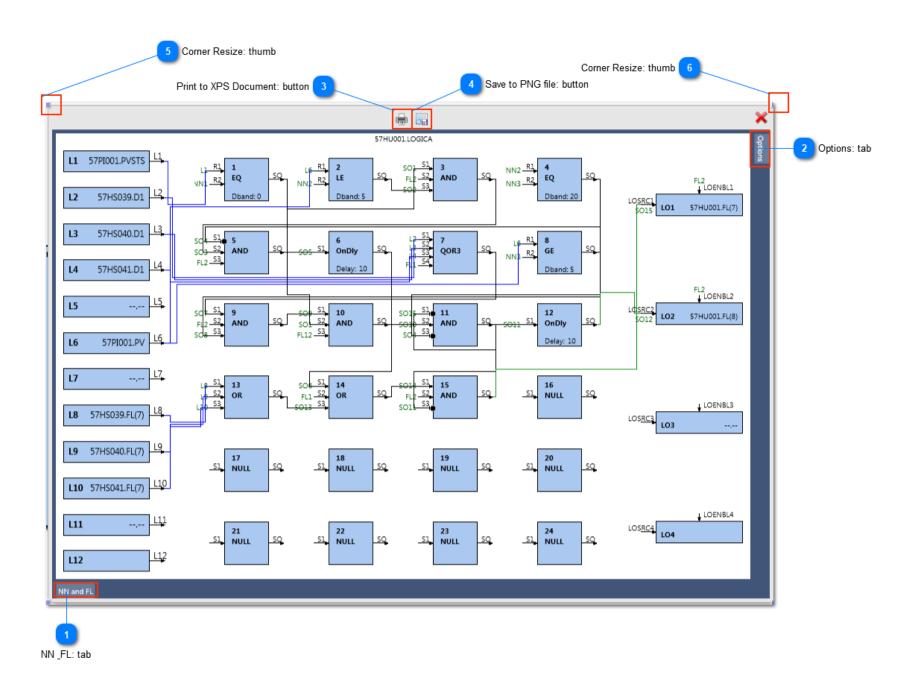
Ctrl+T Use this command within the ChartView to toggle the visibility of all External Reference Tag - shape items.

- This button when clicked copies the Chart View rendering to the Windows Clipboard for pasting into an image application such as Paint.
- This button when clicked displays the properties for a function block or module.
- This mouse target provides a ToolTip that displays the external references attached to a parameter connection.
- This button when clicked displays all external references to the function block. Note: when hovering over this button, a ToolTip will display the references instead of a view.
- f. This button when clicked displays all expressions for calculation type function blocks. Note: when hovering over this button, a ToolTip will display the expressions instead of a view.
- This button when clicked displays the source code viewer for the CAB block.

# **XPM Logic View Overview**

The XPM Logic View provides a ChartView rendering of the Logic gates, inputs, and outputs that are configured for an XPM Logic point type.

Logic Numeric and Flag variables are visible in a floating Tool Window. User options are also configurable through the "Options" Tool Window. Highlight effects allow you to visually identify connected logic gate relationships to other gates, inputs, or outputs.



NN & FL: tab

Hover your mouse over this tab to view the configured values for Numeric and Flag type variables.

Options: tab



Hover your mouse over this tab to view and/or modify the options for the Logic View.

Print to XPS Document: button

Click this button to print the Logic View to an XPS document file.

Save to PNG file: button

Click this button to save the Logic View to a PNG type image file.

Corner Resize: thumb

Click this resize thumb with your left mouse button and hold to drag the Logic View to a new size. The views elements will also resize when dragging to a new size.



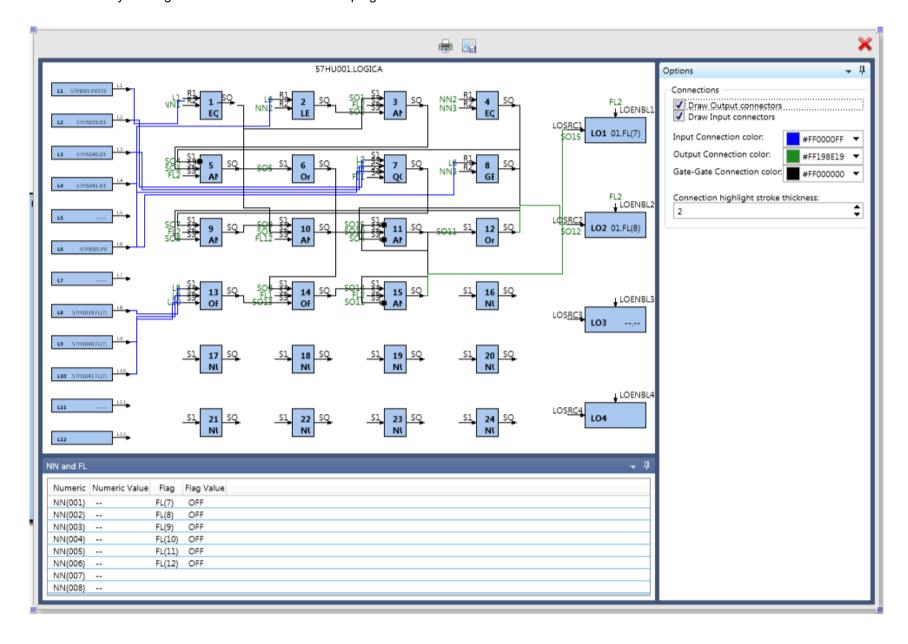
# **Corner Resize: thumb**



Click this resize thumb with your left mouse button and hold to drag the Logic View to a new size. The views elements will also resize when dragging to a new size.

# **XPM Logic Tool windows**

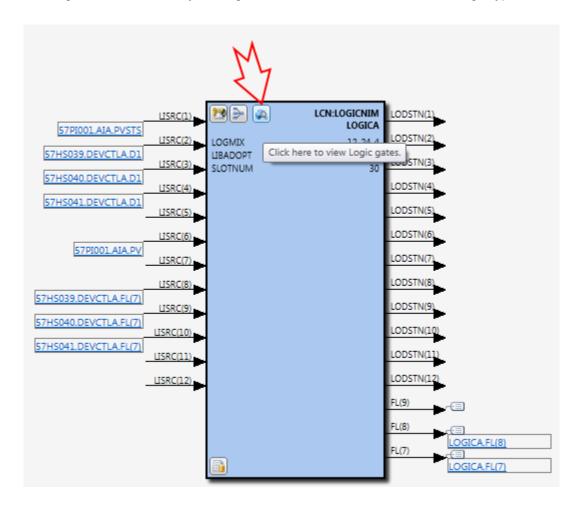
The Logic View contains Tool windows that can be accessed by hovering your mouse cursor over its tab. You can dock the Tool windows as shown below by clicking its Pin button located at the top right of the Tool window.



# **XPM Logic View**

# **Using the Logic View**

The Logic View is invoked by clicking a button on the Adorner of an XPM Logic type Function Block.



The Logic View provides Mouse-Over highlighting effects that allow you to visually identify connected logic gate relationships to other gates, inputs, or outputs. Hover your mouse over a block or connection line to highlight relationships for the connection or block.

The Logic View uses the block color of the Logic Function block for its gate, input, and output block foreground colors. Connection colors are configurable through the "Options" Tool Window.

You can select a block in the Logic View by clicking on its surface. This will render a border to show the selection visual.

You can resize the Logic View by grabbing the views corner thumbs using your left mouse button and while holding down the button drag to a new size. The view will often have to be resized based on its configured LOGMIX value.

#### **Printing the Logic View**

You can print the Logic View to an XPS document or save the Logic View to a PNG image file using the Button targets located at the top of the view.

## **Choosing Connection Colors**

You can choose the color to be used for input, gate-to-gate, and output type connections through the "Options" Tool Window. The default is the color Blue for connections to LISRC(n), the color Black for connections between logic gates, and the color Green for connection to output LOSRC(n) or LOENBL(n) parameters.

#### **Choosing Connection type rendering**

You can choose whether or not input or output type connections are rendered in the view through the "Options" Tool Window.

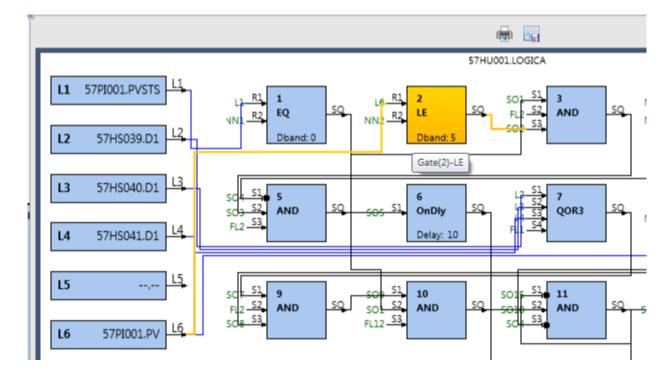
#### **Select the Highlight Connection stroke width**

You can change the thickness of a highlighted connection line through the "Options" Tool Window. The default is a stroke width of 2 units.

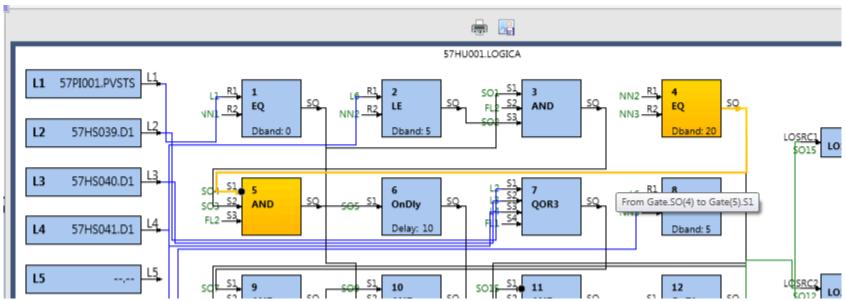
#### **Highlighted Effect examples**

This example shows the highlight effects when the mouse is hovered over a logic gate.

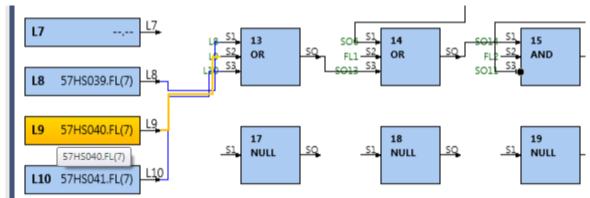
Copyright © 2024 by Automation Software Technology, Inc. All Rights Reserved.



This example shows the highlight effects when the mouse is hovered over a connection line.

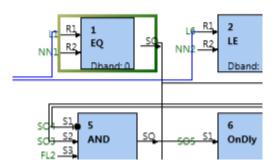


This example shows the highlight effects when the mouse is hovered over an input or output block.

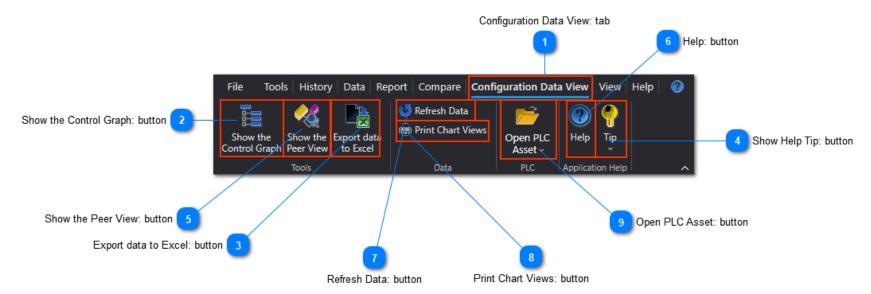


#### **Block Selection**

To select a block in the Logic View click on its surface to render a selection border around the block.



#### ChartView Ribbon menu bar



# Configuration Data View: tab

Configuration Data View

This Toolbar tab becomes visible when the Data View tab is focused. If this tab is not visible when viewing configuration in the Data View, then click on any item in the Data View to focus it, which will then make this Toolbar tab visible.

Show the Control Graph: button



Click this button to view the **Control Graph** for the selected item(s) in the ListView.

Export data to Excel: button



Click this button to view the <u>Data Export</u> wizard for exporting parameter data for the selected item(s) in the data ListView to Microsoft Excel.

Show Help Tip: button



Click this button to view a quick Help Tip related to functions associated with this Data View such as keyboard shortcuts.

Show the Peer View: button



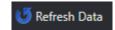
Click this button to show the <u>Peer Map</u> for all Controller and PCDI nodes contained in the configuration project.

Help: button



Click this button to view the help documentation topic for this Data View.

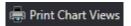
#### 📻 Refresh Data: button



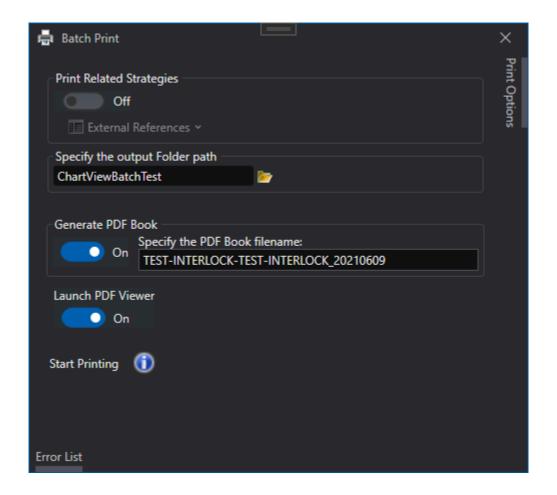
Click this button to refresh the configuration views with any changes that were made to the assets such as importing additional data. Any currently opened Chart Views will be closed prior to refreshing the views.



#### **Print Chart Views: button**



Click this button to print configuration Chart Views. A Print View is displayed that provides options for printing and generating PDF documents.





#### **Open PLC Asset: button**



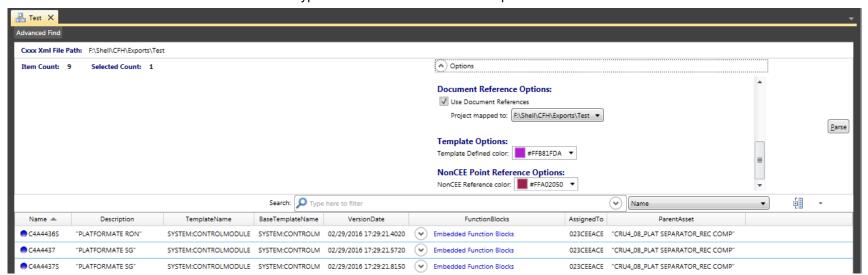
Click this drop-down arrow to see a list of currently imported PLC assets and view their configuration content. PLC assets such as <a href="Triconex platforms">Triconex platforms</a> and <a href="Honeywell FSC platform">Honeywell FSC platform</a> are supported for importing into your configuration projects. Select a PLC Asset from the list to view its configuration and any relationships to existing Control strategies.

#### **NonCEE References**

Chart views provide visual cues that indicate when NonCEE type Point.Parameter references are being used by applying a user defined color to both Watermark indicators on a Function Block surface and also any Pin Connection lines that are drawn in the Chart View. NonCEE type references can be from LCN, SCADA, or other 3rd party data references.

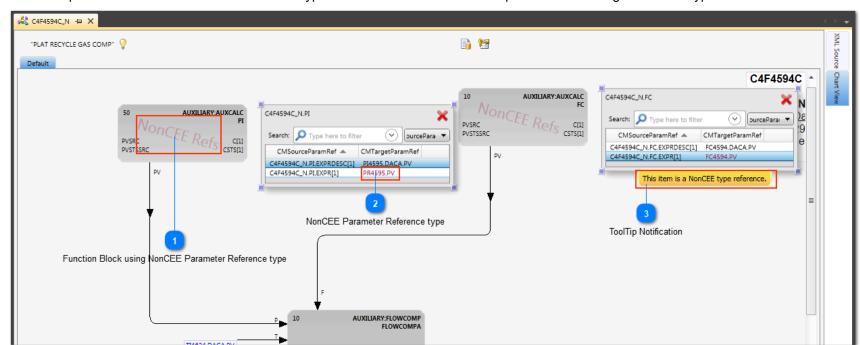
# Choosing the Color to use for identifying NonCEE Point references

The color to choose for NonCEE Point reference types is defined in the Data View options.



### **Examples**

This example shows a Watermark on an AuxCalc type Function Block that has an expression containing a NonCEE type Parameter reference.



Function Block using NonCEE Parameter Reference type



This Watermark in the user defined color is shown on any Function Block surface that references a NonCEE Parameter reference type.

NonCEE Parameter Reference type PR4595.PV

NonCEE Parameter reference types are always shown in the user defined color for NonCEE types.

ToolTip Notification

This item is a NonCEE type reference.

If a Parameter reference is a NonCEE type, then a mouse cursor ToolTip indicates this status.

# Using PolicyCop to find all NonCEE Reference types

Copyright © 2024 by Automation Software Technology, Inc. All Rights Reserved.

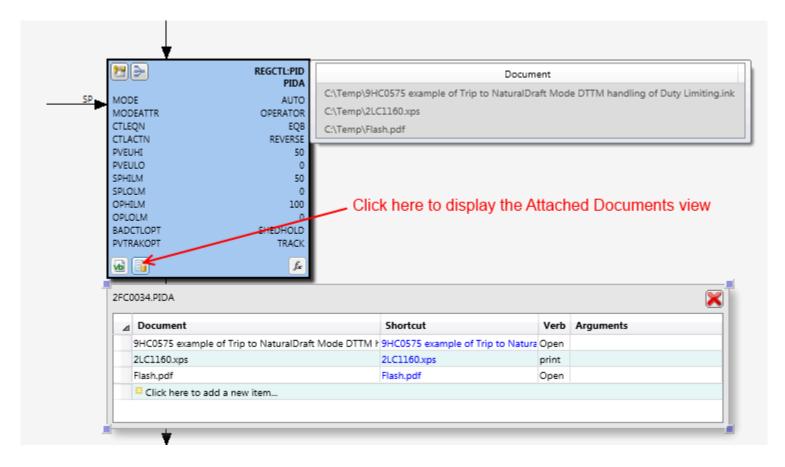
PolicyCop supports an API Method named DataMethods.GetNonCeeReferences() that returns a string array of all CM.BlockName instances where a NonCEE Point reference is being used. There is a Rule provided with the PolicyCop RuleSets named AnyNonCEEReferences that uses this API method that can be executed on any PKS CEE XML configuration project to quickly identify usage.

#### **Document References**

Control Architect provides a feature that allows associating file based documents with a control strategy function block or Control Module.

#### **Attaching a Document Reference**

Document references are attached to ChartView function blocks and the Control Module itself. The following image shows the button target that is clickable to display the document attachment view.



To attach a new Document reference, click the item labeled "Click here to add a new item...". A browse folder image will be displayed in the Document cell of the new row that will allow you to click and browse for the file based reference.

#### **Document Reference Properties**

When a new document is attached it will have its Verb set to *Open* by default. You can choose to change the action that is applied when the hyper link shortcut is clicked such as *Print*. The set of available verbs that are presented in the drop-down list are the verbs that are currently registered for the file extension type on your system.

If the document reference shortcut requires you to specify additional arguments to be passed, you can specify these in the column labeled *Arguments*.

# **Deleting Document References**

To delete an existing Document reference, select the data grid row that contains the reference and press the [Delete] key. You will be prompted to confirm deletion of the item.

**Additional References** 

Project Data Map

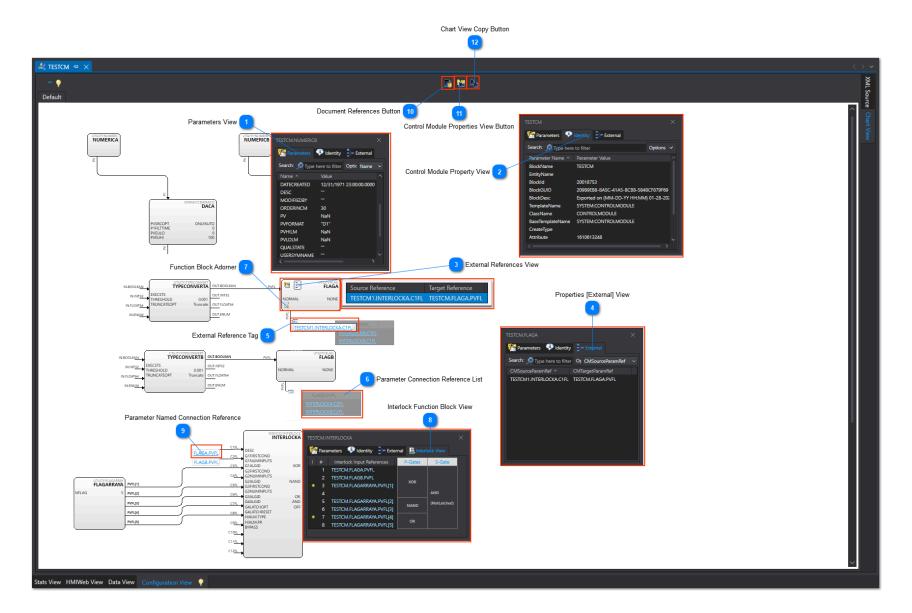
# **Working with Chart Views**

# **Property Views**

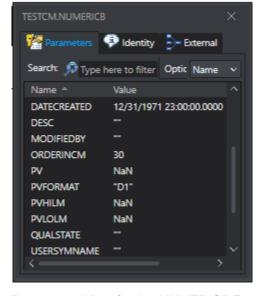
A Chart View provides a visual representation of a Control Strategy using various shapes where the most common shape is the "Function Block". A Function block shape provides the visual representation of the Control Function Block itself along with its full Parameter set that defines the configuration and behavior of that particular Control algorithm.

Control Architect often refers to the Parameters that are configurable on a Function Block as a Block's 'Properties'. Most shapes that are rendered in a Chart View provide a means of viewing its Block Properties "Parameters". When hovering over a Function block presentation in a Chart View an Adorner layer of available function views becomes visible and left-mouse clickable for viewing various aspects of a block. Every Function block that makes up a Control Strategy will always provide several views including Properties, Identification, and External References. Some additional block types may present additional views such as XPM Logic & Device Control types, CEE Interlock, and PLC Block types.

Below are several examples of typical Property views that are available for Function blocks. These Property views were displayed by left-mouse clicking a target button in the Adorner layer that is presented when you hover your mouse over the surface of a Block in the Chart View.



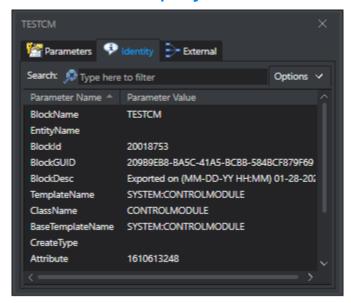
Parameters View



Parameters View for the *NUMERICB* Function Block. This *Parameters* view can be sorted, filtered and copied using Ctrl+C / Ctrl+V keys.



# **Control Module Property View**



Left mouse double-clicking in the background of the Chart View will display the Property View for the Control Module that contains the embedded Function blocks. The Identity tab is selected for this view that shows typical Class and Template information for the block.

# 3

#### **External References View**

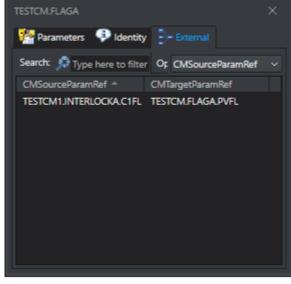


Hovering over the "External References" adorner button will display a ToolTip of any external Parameter references to this particular Function block. Left mouse-clicking the same adorner button will invoke this External References view as shown. These connections are navigable hyperlink references that can be clicked to follow the reference.

Pressing the Keys Ctrl + R will toggle the visibility of all External Reference views in the Chart View.

# 4

#### **Properties [External] View**



Viewing this [External] tab item in the Properties view will also show the list of all external Parameter references for this particular Function block. These connections are navigable hyperlink references that can be clicked to follow the reference.



#### **External Reference Tag**



Block pins that have referenced Parameter connections that reside in another Control Module will display a 'Tag' type shape that when clicked will show Name Parameter Reference Connection. These connections are navigable hyperlink references that can be clicked to follow the reference.

Pressing the Keys Ctrl + T will toggle the expansion of all External Reference tag shapes in the Chart View.



#### **Parameter Connection Reference List**



When a particular Pin connection on a Function block contains multiple Parameter references, then it will be displayed as a "Reference List" that is navigable when clicking the Parameter reference items.



#### **Function Block Adorner**



The Function Block Adorner layer is displayed when you hover your mouse over the block surface. This is the typical set of buttons available for function blocks, but other such as "Expressions", "Code", "XPM Logic" views, etc. may also be visible depending on the Function block type.

8

#### **Interlock Function Block View**



The Interlock View tab item when selected in the Properties View will present a visualization of its configured gate logic. The LED status item in the first column represents the inversion status of the input.



#### **Parameter Named Connection Reference**



Provides a navigable reference to this blocks Parameter C1FL.

10

#### **Document References Button**



Click this button to add/delete/view document references associated with this Control Module instance.

11

#### **Control Module Properties View Button**



Click this button to view the Properties for this Control Module.

12

#### **Chart View Copy Button**



Click this button to copy the Chart View content as a PNG file to the Windows clipboard. Use Ctrl+V to paste this image into another document.

#### **HMIWeb View**

#### **Overview**

The HMIWeb view provides a list of Process Graphic files that have been parsed to show their data references and embedded scripts. When references are parsed from their Process Graphic files, the references are stored into an internal data Model that are integrated into the projects Configuration Asset data Model so that graphic references show as external references to configuration entities. The structure of the HMIWeb Process Graphic can be viewed in a Browser view.

#### **DSA References**

You can provide additional configuration Project references to resolve DSA references. The graphic list will be revalidated against the list of project references to resolve any entity references that are marked as invalid. If an entity reference is found in a project reference, then its invalid state will be cleared.

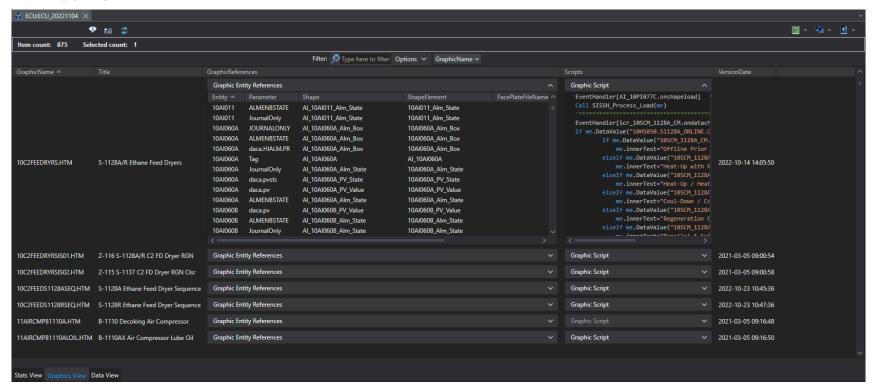
#### Reporting

You can generate reports for viewing in the Report View. The Report viewer allows you to print or export a generated Report book to media types including Adobe PDF, Microsoft Word, Microsoft PowerPoint. You can generate a Report book that contains:

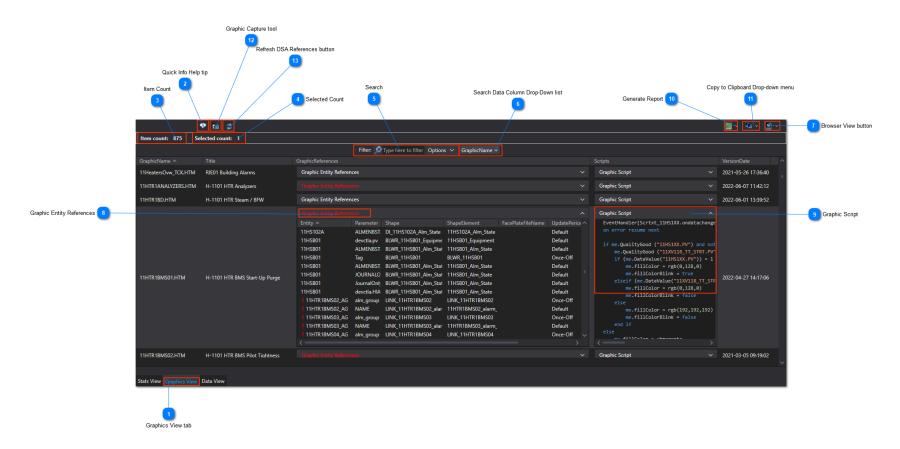
- · Cross reference of Entity usage within graphics
- Detailed process Controller usage within graphics
- · Graphic details including Parameter reference within graphic shapes
- · Invalid entity references in shapes and script
- Entity.Parameter UpdatePeriod counts within graphics

The HMIWeb View also supports copying graphic data to the Windows Clipboard for pasting into Excel.

#### **HMIWeb View**



### **Using the HMIWeb View**



Graphics View tab

Graphics View

Choose this tab to parse and view HMIWeb Process Graphic file references to Cxxx/ACE type entities.

Quick Info Help tip



Hover your mouse cursor over this tip image to view quick help information.

Item Count

Item count: 875

Displays the total count of Process Graphic items in the view.

Selected Count

Selected count: 1

Displays the count of selected Process Graphic items in the view.

🦱 Search



Specify search text to filter the HMIWeb view. The search text specified will be searched upon in the column of data specified in the drop-down list to the right of the search box. Special wild card <u>characters</u> are supported for filtering this view.

Search Data Column Drop-Down list

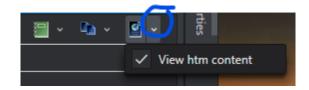


Select the data column to use when searching for text.

Browser View button



Click this button to view any attached Process graphic images { png, pdf } or optionally the Process Graphic static structure in a Browser view. This view can be resized and its location moved. Its last position and size is saved and restored in a new session. While the Browser view is visible, you can change the selected item in the ListView to automatically browse to the newly selected Process Graphic item. By default the viewer will try to find any attached graphic images that represent the Process graphic. If you choose to view the static structure, then click the drop-down button and choose the option *View htm content*.



# Graphic Entity References

Toggle this Expander button to show/hide Entity references. You can hover your mouse cursor over this Expander button to view a ToolTip of references.



Note that when an invalid reference is detected in either a shape or script, it will show the Entity reference in the list with an invalid indicator image and will also change this label text color to red.

Those references that may be DSA types can be resolved by adding additional configuration snapshots as DSA references in the *Data View* tab. When updating the DSA references list you will be required to refresh the validation for the graphics. Use the <u>Refresh</u> button located in the toolbar to re-analyze the graphics for invalid references using the updated DSA projects list.

You can specify <u>additional projects</u> to resolve DSA type references. Script references are shown as invalid if an *Entity.Parameter* reference is not defined in the shapes Script Data holder.

# Graph

#### **Graphic Script**

```
Graphic Script

EventHandler[Scrtxt_11HS1XX.ondatachange on error resume next

if me.QualityGood ("11HS1XX.PV") and not me.QualityGood ("11XV118_TT_STRT.PV" if (me.DataValue("11HS1XX.PV")) = 1 me.fillColor = rgb(0,128,0) me.fillColorBlink = true elseif (me.DataValue("11XV118_TT_STR me.fillColor = rgb(0,128,0)
```

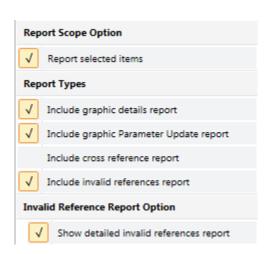
Toggle this Expander button to show/hide scripts. You can hover your mouse cursor over this Expander button to view a ToolTip of scripts found for embedded shapes in the Process Graphic file.

# 10

## **Generate Report**



Click the drop-down arrow to display its menu. These menu commands generate reports that can be viewed, printed and exported to various media types including Adobe PDF, PowerPoint, Word, etc.



Tip: The "Include cross reference report" option if used while the option to report on only selected items is chosen will still require iterating through all Process graphics as external references are not specific to a graphic.

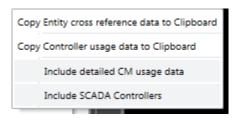
Tip: The "Include graphic details report" option if used can take a considerable amount of time to generate and may include thousands of pages of graphic details. Use this option with caution on an entire HMIWeb graphics project. Closing the Report View or one of its Report tabs will automatically cancel any active Report generation task.

# 11

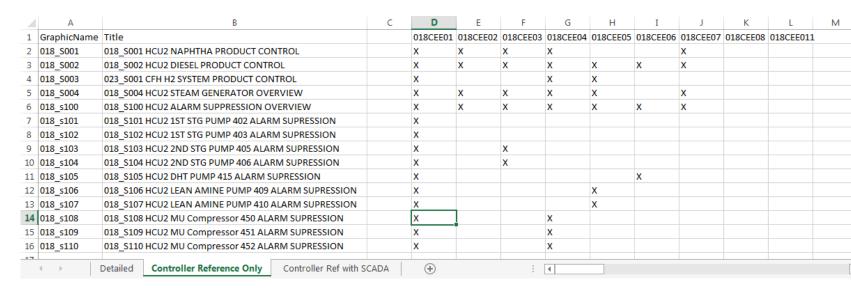
## Copy to Clipboard Drop-down menu



Click the drop-down arrow to display its menu. These menu commands copy graphics data to the Window Clipboard for pasting into Excel. A HMIWeb graphics, process controller usage, and graphic details.



This function is often used to provide Controller usage vs. graphics to help define FAT and startup scopes.



You can optionally choose to include detailed data together with the AssignedTo Controller usage data. Detailed data would consist of the Enusage.

You can optionally choose to include SCADA type controllers in the list of Controller names.



## **Graphic Capture tool**



Click this button to launch the Graphics Capture tool.

This tool programmatically launches the Honeywell HMIWeb Graphics editor installed on your local system where Control Architect resides. It will process a selected set of HMIWeb Process graphic files opening them in the HMIWeb view editor while taking a static snapshot of the rendered image and converting the rendered image to either a PDF or PNG file type. These captured images can then be imported into your configuration asset snapshot and viewed in this Graphics view for a selected graphic item.

It is required that HMIWeb Graphics editor is installed on your local system. Honeywell provides a standalone installation for this application that allows you to install the editor on your local desktop or laptop.



#### **Refresh DSA References button**

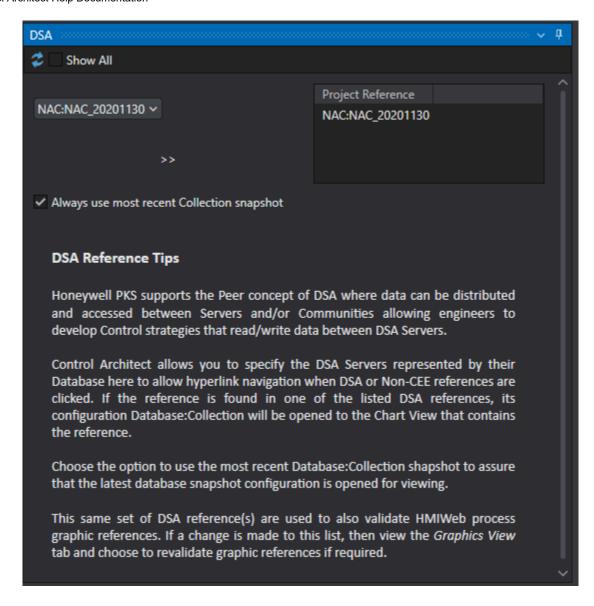


The configuration views *Data View* tab allows you to specify additional configuration snapshots that may contain other DSA references in your Process graphic files.



Click the DSA Toolbar tab to view the DSA references view.

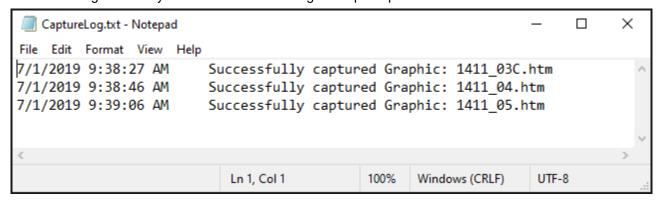
This DSA References view allows you to choose other configuration snapshots to provide as DSA reference resolution when analyzing Process graphics.



# **HMIWeb Graphic Capture**

The HMIWeb Graphic Capture tool is designed to convert HMIWeb Display Builder graphic views into either PDF or PNG content files. The generated PDF or PNG image files are typically used to document the graphic content and provide an associated image when importing HMIWeb Graphic content into a Control Architect configuration snapshot.

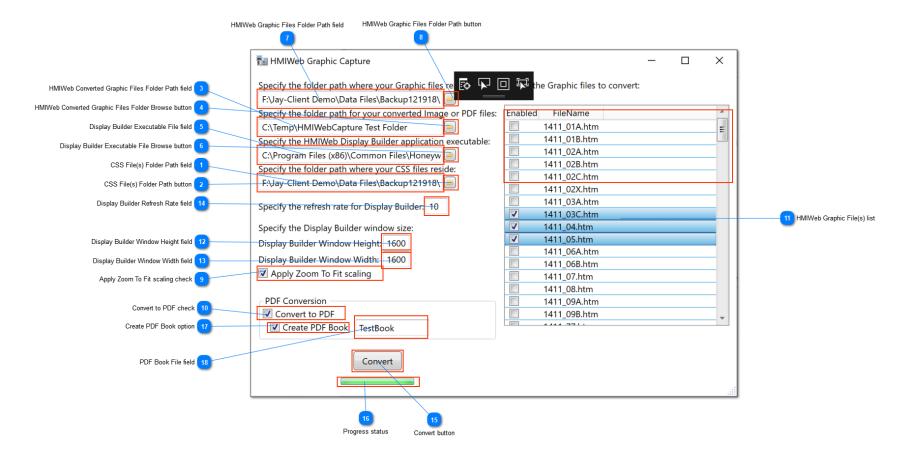
The capture process is logged into a text file named "CaptureLog.txt" that is located in the same folder as your captured documents. Please refer to this log file for any issues discovered during the capture process.



#### **Capture Tips**

- Performance may vary on hosted machines that may require adjustment of default refresh rate of 10 seconds. Increase this if more time is needed to process higher density graphics.
- Do not allow other windows to overlap the launched Display Builder view during the capture execution.
- Adjust the Display Builder View width and height to produce the desired capture document.

# **HMIWeb Graphic Capture Overview**



CSS File(s) Folder Path field

F:\Jay-Client Demo\Data Files\Backup121918\

Click the Browse button to locate the folder path where your HMIWeb Graphic CSS files reside.

CSS File(s) Folder Path button



Click this Browse button to locate the folder path where your HMIWeb Graphic CSS files reside.

HMIWeb Converted Graphic Files Folder Path field

C:\Temp\HMIWebCapture Test Folder

Click the Browse button to locate the folder path where your captured/converted HMIWeb Graphic PNG/PDF content files will be generated to.

MIWeb Converted Graphic Files Folder Browse button



Click this Browse button to locate the folder path where your captured/converted HMIWeb Graphic PNG/PDF content files will be generated to.

Display Builder Executable File field

C:\Program Files (x86)\Common Files\Honeyw

Click the Browse button to locate the HMIWeb Display Builder executable file for launching during the capture/convert operation.

This file is typically located in:

C:\Program Files (x86)\Common Files\Honeywell\HMIWeb\HDB.EXE

Display Builder Executable File Browse button



Click this Browse button to locate the HMIWeb Display Builder executable file for launching during the capture/convert operation.

This file is typically located in:

C:\Program Files (x86)\Common Files\Honeywell\HMIWeb\HDB.EXE

HMIWeb Graphic Files Folder Path field

F:\Jay-Client Demo\Data Files\Backup121918\

Click the Browse button to locate the folder path where your HMIWeb Graphic source files reside.

Copyright © 2024 by Automation Software Technology, Inc. All Rights Reserved.

8

## **HMIWeb Graphic Files Folder Path button**

Click this Browse button to locate the folder path where your HMIWeb Graphic source files reside.

# Apply Zoom To Fit scaling check

Apply Zoom To Fit scaling

Choose this option to set the Display Builder "Zoom to Fit" option prior to capturing and converting the graphic content.

## Convert to PDF check

✓ Convert to PDF

Choose this option to generate PDF content vs PNG content. If this option is not checked, then graphic content will be generated as PNG image files.

# HMIWeb Graphic File(s) list

Enable	d FileName	^
	1411_01A.htm	
	1411_01B.htm	
	1411_02A.htm	
	1411_02B.htm	
	1411_02C.htm	

Displays the list of HMIWeb Graphic files that were found in the specified folder path. Check the graphic files to capture/convert.

# 👝 Display Builder Window Height field

1600

Specify the Display Builder Window Height to set prior to capturing and converting the graphic content.

## Display Builder Window Width field

1600

Specify the Display Builder Window Width to set prior to capturing and converting the graphic content.

#### 🛖 Display Builder Refresh Rate field

10

Specify how long to delay in seconds before taking a capture of the Display Builder graphic content. The default value of 10 seconds should work for most installations.

#### 🥋 Convert button

Convert

Click this button to begin capturing and converting the enabled HMIWeb graphic files in the list.

# Progress status

Indicates the current progress status of the capture/convert operation.

#### Create PDF Book option

Create PDF Book

Choose this option to combine generated PDF pages into a single PDF Book file. Embedded PDF page content is navigable through generated Bookmarks.

#### 😱 PDF Book File field

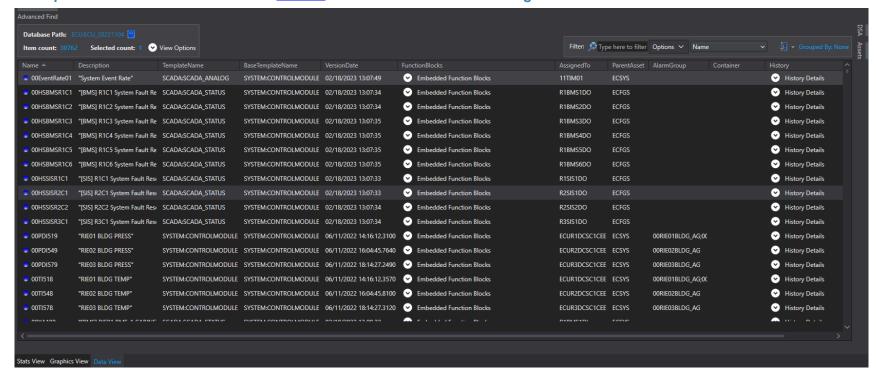
TestBook

Specify the name of the PDF Book file that will be created to contain the captured PDF document pages.

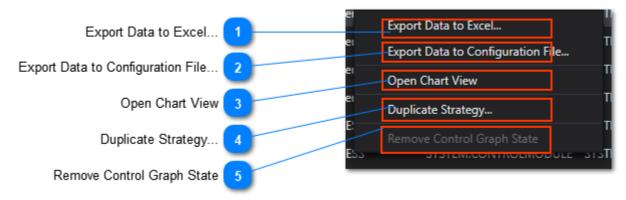
#### **DataView**

This view provides a list of Control Module entities that reside in a Configuration project that was generated by importing PKS Server exported configuration that includes CDA (C300), SCADA, and LCN exported EB (Exception Build) files.

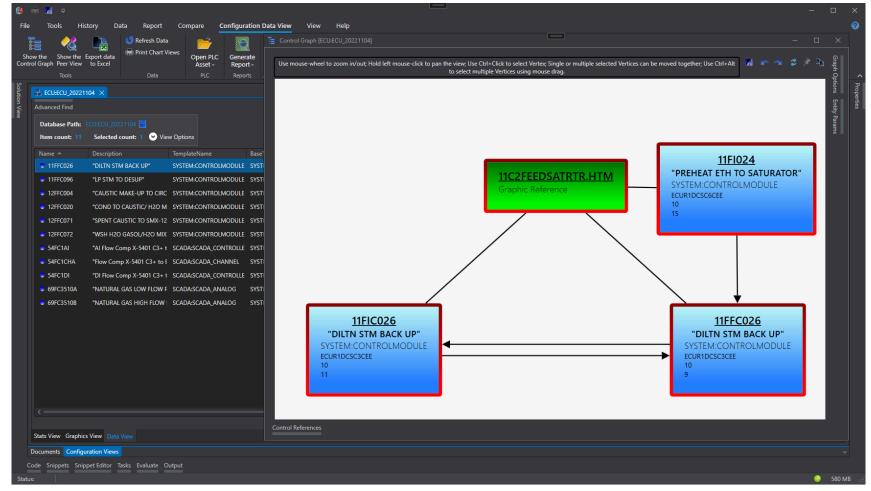
#### Example of a DataView that has been filtered to remove LCN entities using "!LCN".



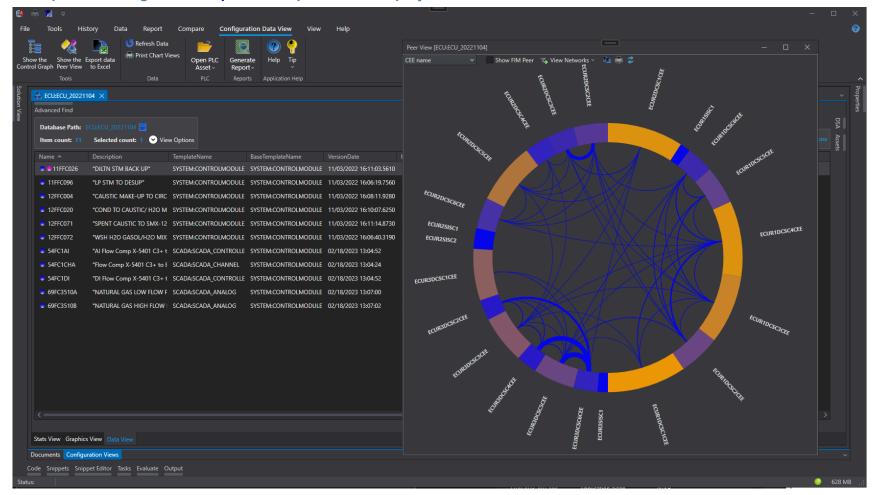
#### Data View right mouse-click menu options.



## Example of a **Control Graph** invoked from a single selected Control Module entity.



#### Example of viewing a **Peer Map** for an Experion Server project.



#### **Tips & Tricks**

- Left mouse double-click on a Control Module item in the list will automatically display its visual ChartView.
- You can change the entity selection in the list while the Control Graph view is visible to change the analysis results.
- Pressing the key combination Ctrl+R will copy the selected item(s) external function block references to the Windows clipboard for pasting into Excel.
- Pressing the key combination Ctrl+C will copy the selected item(s) for pasting into Excel.
- Pressing the key combination Ctrl+Shift+C will copy the selected item(s) in a format where the embedded function blocks occupy individual rows for pasting into Excel.
- Hovering over the image located to the left of the Control Module name will display a ToolTip list of external references to the Control Module
- Pressing the key combination Ctrl+Shift while left mouse-clicking an SCM sequence block will automatically navigate to its SCM# code block in the SCM# view.

# Export Data to Excel... Export Data to Excel...

Choose this menu option to export configuration data to Excel using the Export Wizard tool.

# Export Data to Configuration File... Export Data to Configuration Fi

Choose this option to export the configuration to its original PKS formatted XML and BCD configuration files. You can use this option to generate valid configuration files from data that was imported into the Document server. These exported configuration files can then be imported back into PKS Control Builder. This option is only valid for PKS CDA type configuration.

# Open Chart View Open Chart View

Choose this option to open the selected Entity item to render its Chart View.

# Duplicate Strategy... Duplicate Strategy...

Choose this option to duplicate the configuration for the selected Entity items. This function allows you to quickly build configuration based on existing Entity configuration. Parameter changes can be made to the duplicated strategies prior to building.

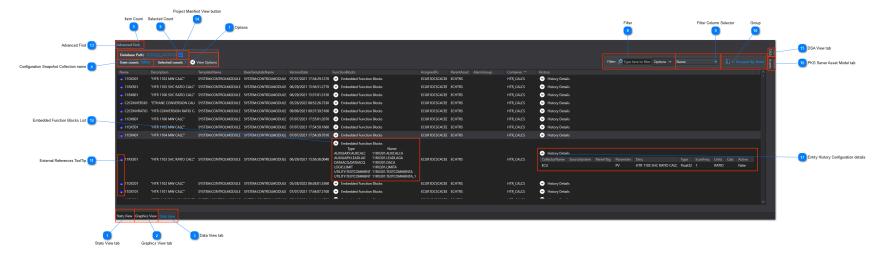


# **Remove Control Graph State**

Remove Control Graph State

Choose this option to remove the saved Control Graphic visual state for this selected Data View item.

#### **DataView Overview**



# Stats View tab



Click this tab to show the Process Controller statistics for this configuration project.

Graphics View tab



Click this tab to view the HMIWeb graphics associated with this configuration project. LCN schematics can optionally be exported from Native Window as debug type "DX" files that are then imported to display graphic references similar to HMIWeb graphics.

Data View tab



Click this tab to view the Data View associated with the current PKS Xml project.

Configuration Snapshot Collection name



Displays the current Database: Collection path in the MongoDB Server database.

Item Count



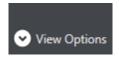
Displays the total number of entities contained in this configuration project Data View based on the current filter criteria.

Selected Count



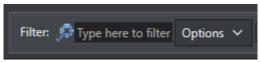
Displays the count of selected items.

Options



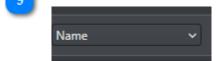
Click to expand/collapse the options view.

Filter



Enter text to filter the Data view. Special wild card characters are supported for filtering the Data view.

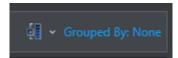
Filter Column Selector



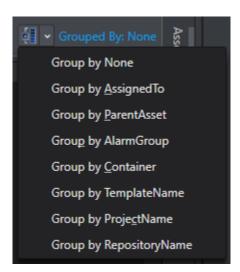
Select a Data view column name from the drop-down list to use when specifying filter criteria.



#### **Group**



Click this button to group the data in the view. You can group the view data on the data columns shown below:

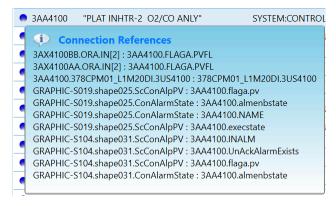


# External References ToolTip



Hover over this image with your mouse to view a ToolTip of external references for the entity item. This blue LED image will be default visible on every item in the list. When the mouse cursor is hovered over the image a query is executed to find all external references for this Entity item. A ToolTip will be displayed that shows any external references that are found.

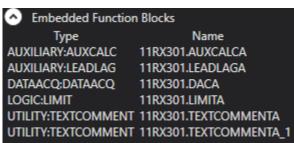
This example below shows the ToolTip that is displayed when the mouse cursor is hovered over the blue LED image for Entity item 3AA4100.



This example shows the ToolTip that is displayed if no external references were found.



#### Embedded Function Blocks List



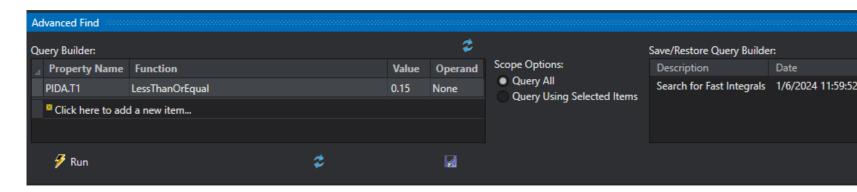
Hover over the text labeled "Embedded Function Blocks" to view a ToolTip list of embedded function blocks for the CM entity or click this button to expand/collapse the item to view the embedded function blocks.

# 13

#### Advanced Find

Advanced Find

Click this tab to show the "Advanced Find" view. This allows you to specify and build complex expressions for filtering the items in the Data v



# 14

#### **Project Manifest View button**



Click this button to show the Manifest for this configuration project. This Manifest view shows the source data that was used to generate this

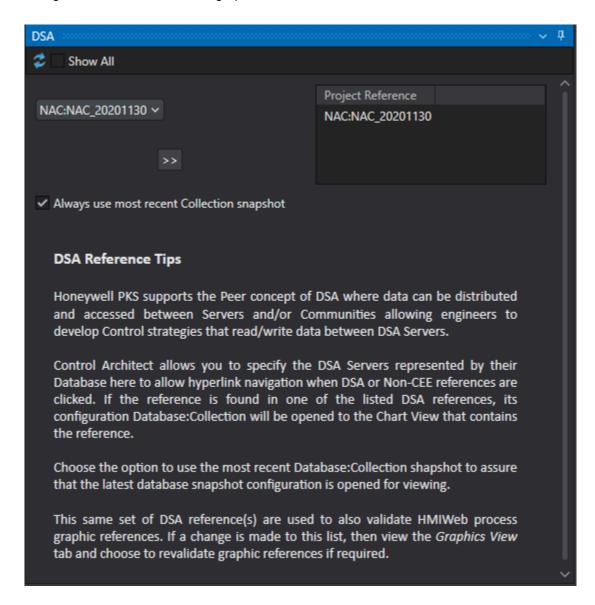


# 15

## **DSA View tab**



To help resolve invalid references contained within Process graphics you can optionally specify other PKS distributed Servers that are acting as DSA references in the graphics.

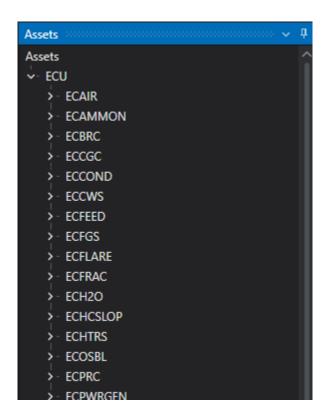




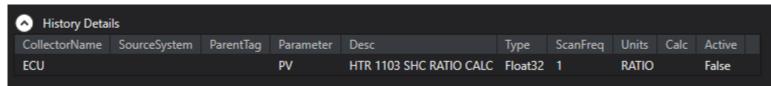
#### **PKS Server Asset Model tab**



When importing configuration assets into a Configuration Collection snapshot you can optionally specify the exported PKS Server Asset model. The import function expects an exported XML data source containing the PKS Asset model.



# Entity History Configuration details



When importing configuration assets into a Configuration Collection snapshot you can optionally specify PHD or PI history exported data source files into your snapshot.

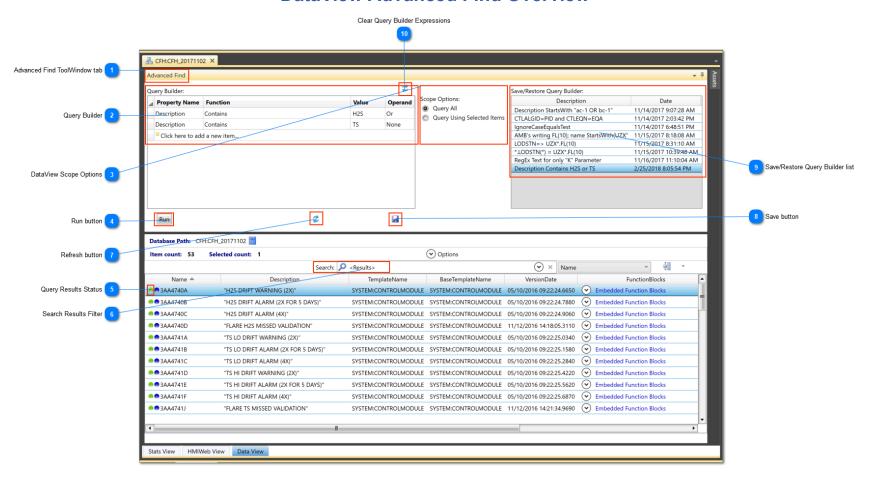
# **DataView Filtering**

Generally a view supports advanced filtering expressions using the wild card character set "!, \*, ?".

Using these special filtering characters allows you to filter the data to a particular subset.

See the topic that explains the use of filtering or searching for items in a ListView type element.

#### **DataView Advanced Find Overview**



The Advanced Find view allows you to define expressions to query data items in the DataView. You can define simple expressions using the predefined functions or you can specify complex RegEx expressions for advanced query needs.

Advanced Find ToolWindow tab

Advanced Find

Click this ToolWindow tab to open the view window. Click the Pin button located in the top-right corner of this view to pin the view in a docked state.

Query Builder



The Query Builder is where you build query expressions by specifying a Property Name, Function, Value, and optionally an Operand. Multiple expressions can be logically combined using the AND or OR operand.

The Property Name that you specify represents the Parameter name of the entity you wish to find.

Wildcard expression syntax is supported when defining the Property Name column value. The following expression for the Property Name column...

"\*PKGNAME\*" indicates to find all entity parameter names that contain the text phrase "PKGNAME". Notice how the '\*' character precedes and ends this specified Parameter name. The wildcard char '\*' means to ignore any characters preceding the name "PKGNAME" and any characters ending after this Parameter name. It will successfully match and find these examples using this syntax:

- UC100.PKGNAME(1)
- UC100.PKGNAME(2)
- UC100.PKGNAME(3)
- ...

# DataView Scope Options

Scope Options:

- Query All
- Query Using Selected Items

Choose the scope option to execute the Query against in the Data View.



#### **Run button**



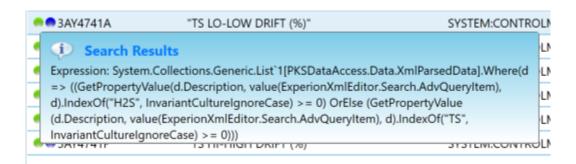
Click this button to execute the defined query expressions against the data located in the DataView.

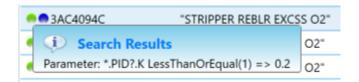


## **Query Results Status**



When a query has been executed against the DataView, the items contained in the Data View represent data that satisfied the query expression. A status LED indicator will show the results of the expression that was executed on a DataView list item.





# 6

## **Search Results Filter**



The Search Filter value will be replaced with the text value "<Results>", which will automatically filter the DataView list for only items that are part of the last executed query result set. Clicking the X button in the Search bar will reset the DataView to its unfiltered set of items.

# 7

#### Refresh button



Click this button to apply the last executed query results to the DataView. This function simply enters the value "<Results>" in the DataView search filter bar.

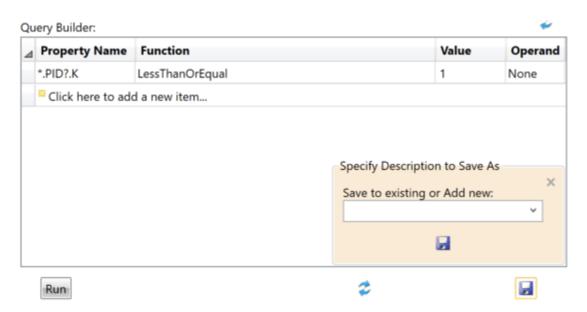
# 8

## Save button



Click this button to save the Query Builder expressions for later restore.

A popup view will become visible that allows you specify how to save the Query Builder expressions.



Specify a description that will uniquely identify the Query Builder expression(s) when saved. Saved Query expressions are added to the "Save/Restore Query Builder" list. Expressions in the saved list can be restored back to the Query Builder by left-mouse double-clicking on an item to restore.

You can also choose to overwrite an existing saved Query item by choosing the item from the drop-down list.

Click the Save button in this popup view to complete the save operation or click the X button to close the popup view.



# Save/Restore Query Builder list

Save/Restore Query Builder:

AM
PM
PM
AM
AM
B AM
4 AM
M

When you save the query expressions contained in the Query Builder they will appear in this list of items for later restore. Left-mouse double-click on an item in this list to restore its saved query expressions to the Query Builder.

You can delete selected item(s) in this list by pressing the keyboard Delete button.

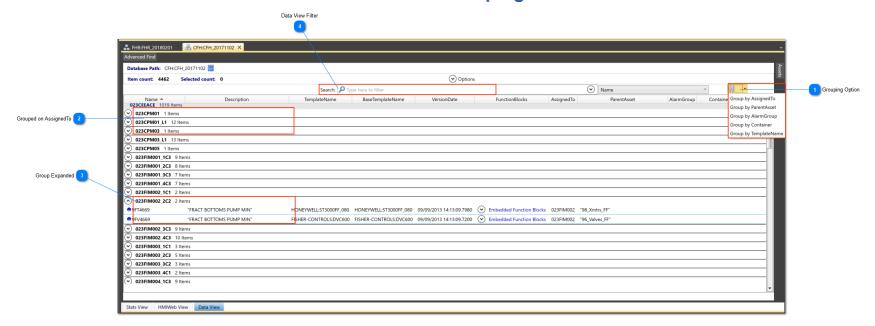


# **Clear Query Builder Expressions**

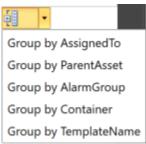


Click this button to quickly remove all existing Query Builder expressions.

#### **DataView Grouping**



# Grouping Option



You can choose to group the DataView based on *AssignedTo* or *ParentAsset* property values. Click the drop-down arrow to display the menu items that allow you to choose the type of grouping to apply. Clicking the Group option button without clicking the drop-down will reset the view to its original flat list.



You can also choose to filter the Group items by specifying a value in the Filter edit box. (i.e. specifying the value of "CEE" as the filter expression would filter the DataView groups such that only *AssignedTo* Group items that contain the text "CEE" would be visible in the view)

# Grouped on AssignedTo



When the group option *AssignedTo* is chosen the DataView is organized into groups where each Group item represents an AssignedTo value.

#### Group Expanded

023FIM002_2C2	2 Items	
9FT4669	"FRACT BOTTOMS PUMP MIN"	н
9FV4669	"FRACT BOTTOMS PUMP MIN"	FI:

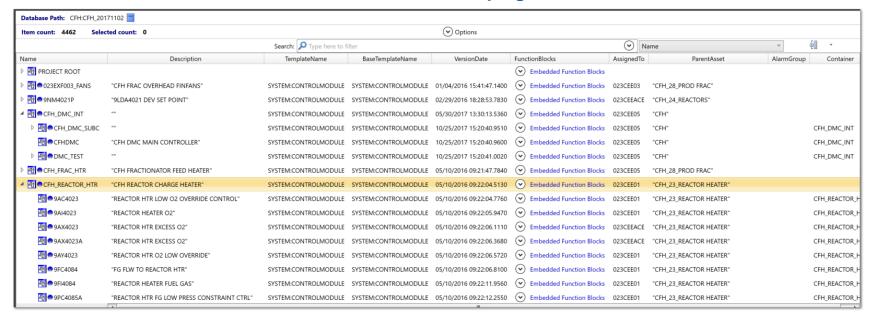
This is the Assigned To Group Item "003CPM04\_L1" expanded to show its contained items that are assigned to this group.

## Data View Filter



Specify additional filtering requirements for the DataView.

# **Container Grouping**



This is an example of grouping on the "Container" property. Control modules that contain other embedded Control modules are shown in their containment hierarchy in a Tree type view.

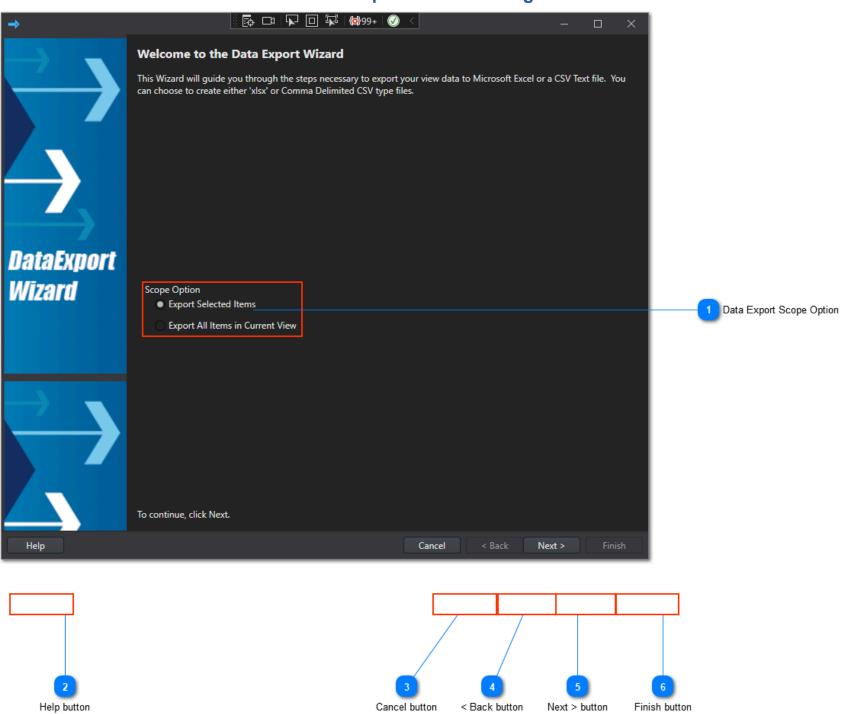
# **Data Export Wizard**

The DataView supports exporting entity parameter data to Excel or a CSV (comma delimited) text file using its Data Export Wizard. Entity parameter scope can be defined along with the column ordering in the exported Excel Workbook file. Excel Worksheet formatting is also supported.

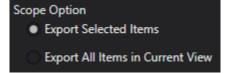
#### **Getting Started**

- 1. From the DataView tab of a PKS Configuration View, select the items to export in the ListView. An option in the Export Wizard allows you to export all items without having to select all items in the ListView.
- 2. In the Scope Option of the Wizard page select the scope of the items to export.
- 3. Choose the type of Export file to create in the Export options.
  - a. Only Export file formats \*xlsx for Microsoft Office 2007+ are supported.
- 4. Click the file browse button to specify the file name to create and its location.
- 5. In the Wizard Parameter Selection page expand the Options expander.
  - a. Choose the Parameter Scope
    - i. Choose CM parameters to export only the parameters and their values visible in the DataView tab.
    - ii. Choose All embedded parameters to export parameters found in embedded Function blocks.
  - b. Choose the Parameter Source
    - i. If All embedded parameters is chosen as the Parameter scope, then you must specify the parameters to export.
      - 1. Click the **Generate** button to generate a list of all parameters found in all embedded Function blocks using the selected items in the DataView ListView as the scope of Control Modules and embedded Function blocks to process.
      - 2. When the generation has completed, the Templates and Parameters lists will contain the parsed Templates and Parameters.
  - c. Click the **Generate** button to generate a list of Templates and Parameters from the selected Control Module items in your Data View.
    - i. If the Export scope from the Data View is for all items, then all Control Module items will be parsed to extract their embedded Function Block Template and Parameter items.
  - d. When a new list of Parameter items is generated all its parsed Templates and Parameter items are not enabled for export. You must enable the Template and Parameter items to use in your export by checking the "Enabled" Checkbox. When an item is enabled it will be visible in the enabled parameter items list.
    - i. You can choose to exclude certain Templates or Parameter items.
    - ii. You can also move Parameter items to the front of the list such as identification type parameters { BlockName, ClassName, etc. }. By default the created Excel Worksheet will order the column items according to the order in this list.
  - e. To save this list of parameters for later recall, click the folder browse drop-list button to the right of the File Path TextBox and select the menu item **Save As**. Specify the File name and location to save the Template and Parameter list to.
    - i. Click the **Save** button to save to the specified file.
- 6. Set Parameter formatting and header styling options.
  - a. Expand the Parameter Formatting and Parameter and Header Styling expanders to view additional formatting options such as Font type, size and color.
- 7. Click the Wizard Next button to begin exporting the selected Parameter items to the specified Excel Workbook file.

# **Data Export Welcome Page**



Data Export Scope Option



Choose to export the selected entity items in the DataView list or all entity items.

Help button

Click this button to view Help in the help documentation for the active Wizard page.

3 Cancel button

Click this button to cancel and close the Wizard view.

Sack button

Click this button to navigate to the prior Wizard page.

Next > button

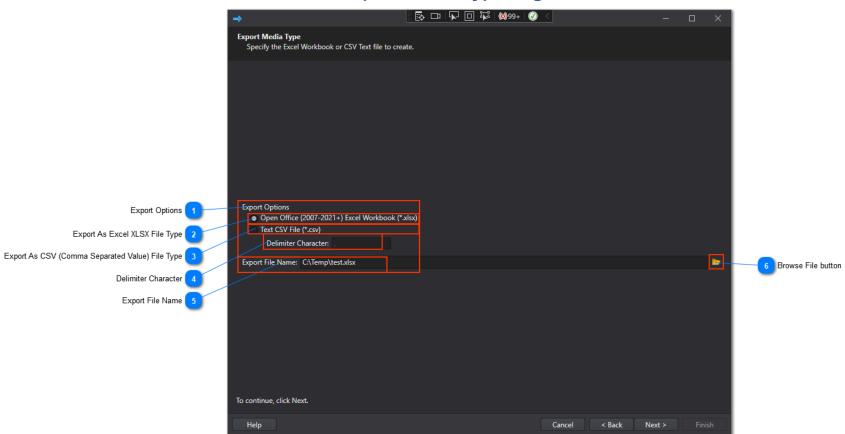
Click this button to navigate to the next Wizard page.

Copyright © 2024 by Automation Software Technology, Inc. All Rights Reserved.

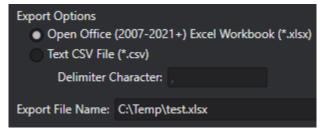


Click this button when the export task has completed to close the Wizard view.

## **Data Export Media Type Page**



# Export Options



Choose the type of export file and data to create.

# Export As Excel XLSX File Type

Open Office (2007-2021+) Excel Workbook (\*.xlsx)

Choose this option to create an XLSX Excel type file that is supported by Microsoft Office 2007+.

# Export As CSV (Comma Separated Value) File Type

Text CSV File (\*.csv)

Choose this option to create an CSV text type file that can be opened in application such as Excel.



Parameter data values that contain the specified delimiter character will have the delimiter value replaced with the character '~'.

#### Delimiter Character

Delimiter Character:

Specify the delimiter character to use for CSV type files. The default character is the comma ','. You can change this delimiter to another character such as the tab '\t' character.



Parameter data values that contain the specified delimiter character will have the delimiter value replaced with the character '~'.

#### Export File Name

Export File Name: C:\Temp\test.xlsx

Specify the file path where the exported data will be exported to. You can use the [**Browse**] button to browse to a folder location and specify the file name.

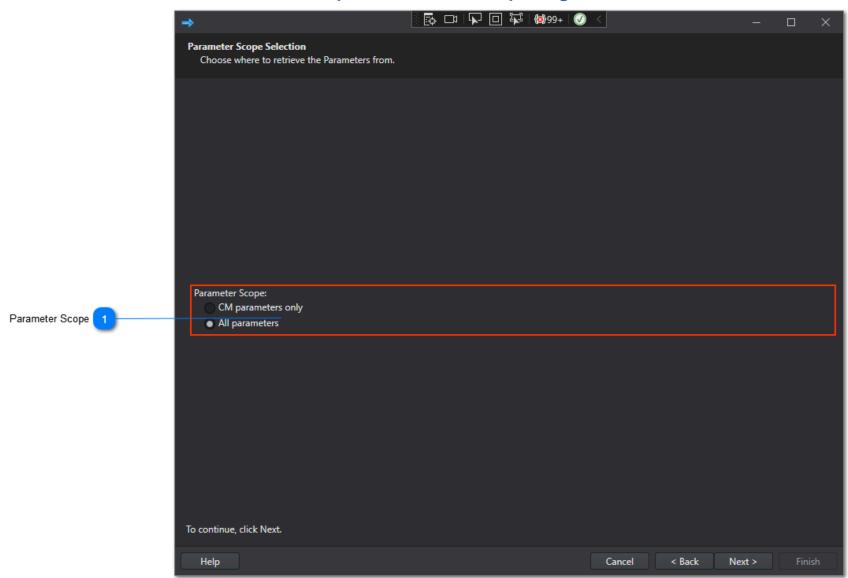
#### Browse File button



Click this button to specify the export file name and its location.

Copyright © 2024 by Automation Software Technology, Inc. All Rights Reserved.

# **Data Export Parameter Scope Page**

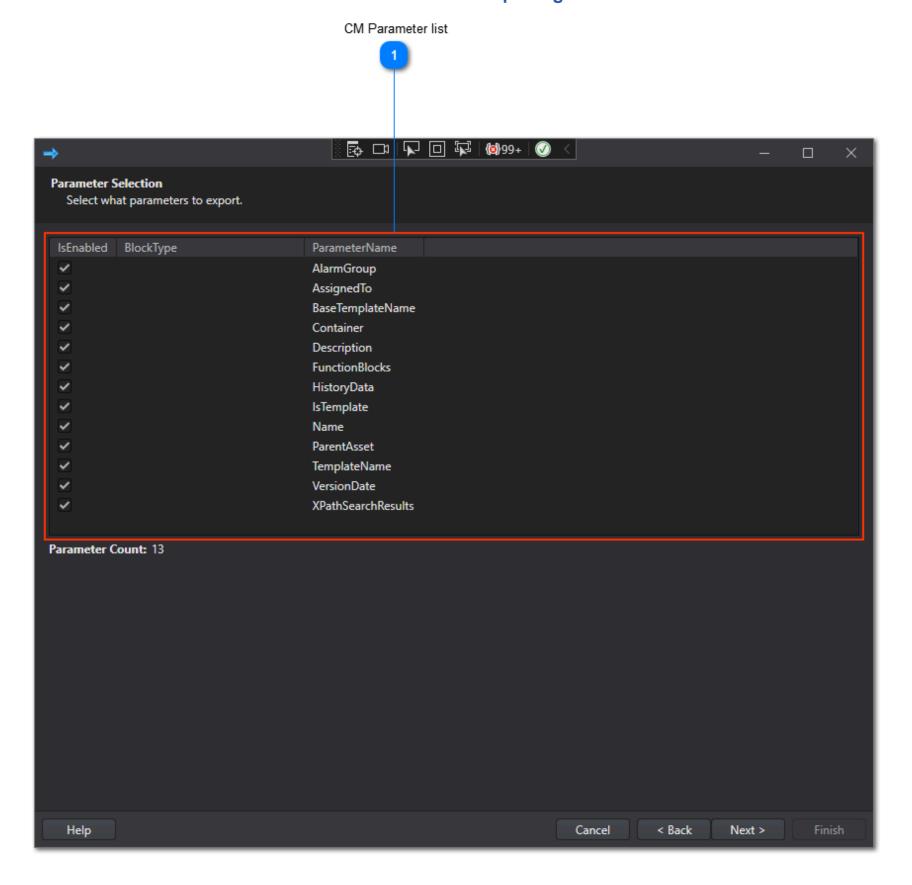


# Parameter Scope

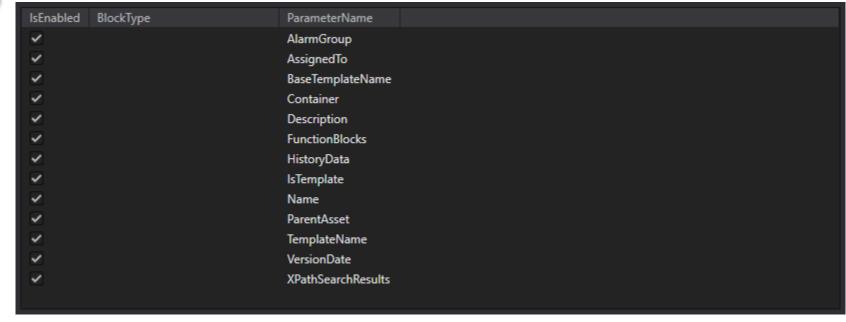


Choose whether to export the common Control Module parameters or embedded Function blocks parameters. To export embedded blocks parameters, you must specify the list of parameter names to export. The 'common' set of CM parameters are those data columns found in the Data View and include { Description, TemplateName, BaseTemplateName, AssignedTo, ParentAsset, AlarmGroup, etc. }

# **CM Parameter Scope Page**



#### **CM Parameter list**

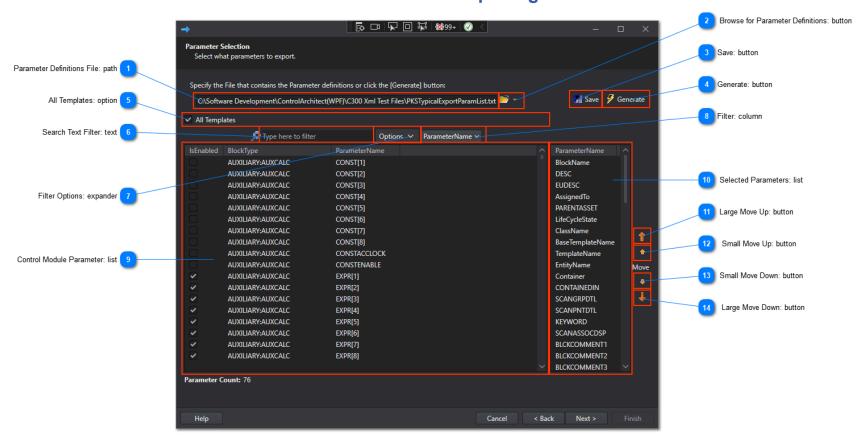


The scope of Parameters that are available to export when choosing the CM Parameter Scope option.

You can choose to enable a certain set of these Parameters to export by checking the items. You cannot change the order in which they appears when exported.



## **All Parameter Scope Page**



# Parameter Definitions File: path

 $C:\Software\ Development\ControlArchitect (WPF)\C300\ Xml\ Test\ Files\PKSTypical Export Param List.txt$ 

Click the drop-down button to select a command to execute on a Parameter definition file.

You can choose to open an existing Parameter definition file or to save the Parameter definitions to another file.

## Browse for Parameter Definitions: button



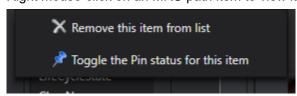
Click the drop-down button to select a command to execute on a Parameter definition file.

You can choose to open an existing Parameter definition file or to save the Parameter definitions to another file.

Click the drop-down button to expose a list of Most Recently Used (MRU) file paths. Click a path to open its set of parameter definitions.



Right mouse-click on an MRU path item to view its context menu that allows you to remove the item or toggle its pinned state.



#### 🤧 Save: button



Click this button to save any changes made to the current Parameter definitions. If this is a new set of Parameter definitions being created, then click the drop-down button and choose the **SaveAs** menu option to save to a new file.

## Generate: button



Click this button to generate a default list of templates and their parameters based on the Data View export scope. If selected items are being exported, then only the selected items will be parsed to extract their embedded templates and parameter items. This list of generated Parameters allows you to pick what Parameters to export.

Copyright © 2024 by Automation Software Technology, Inc. All Rights Reserved.



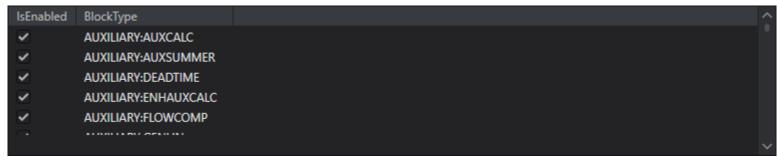
## **All Templates: option**

✓ All Templates

Choose this option to enable all found Templates in the generated scope of Parameters. If you wanted to reduce the size of the generated Parameter list you can choose to disable a specific Template and its Parameters.

This list contains the Template items that is the basis for the Parameter items in the Parameter list. This list is populated when a "Generate" command is executed on the selected Control Module items in your Data View. You can choose to disable a particular Template from being exported by removing the check status in the Template "IsEnabled" column. When a Template column is disabled from exporting, the Template Parameter items in the available Parameters list do not change their enabled status.

Checking this option will auto-hide the list of available Template:Parameter items.



The generated Parameter items are shown in the format Library:Template.



#### Search Text Filter: text



The search text that is used to filter the list of parameters. Using this filter will allow you to quickly focus on the Parameter names of interest.



#### Filter Options: expander



Click this expander button to expand/collapse the filter options pane.

8

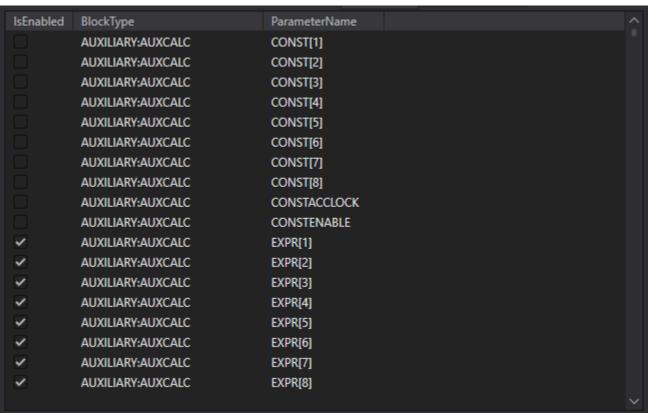
#### Filter: column

ParameterName >

Select the data column from the drop-list to use when specifying filter text.



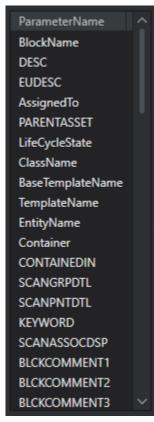
#### **Control Module Parameter: list**



The list of generated Control Module parameters that can be selected for exporting. This set of parameters are based on what Templates have been chosen to use for selecting to export.

10

#### **Selected Parameters: list**



This list contains the Parameter items that have been enabled for inclusion in the generated Excel export file.

You can arrange the order of the items in this list by using the "Move" buttons to move an item or items up or down in the list. The order in this list is used when generating the Excel column names and position in the Worksheet.

Tip: Use the Parameter name "BlockName" to export the name of the Entity that is being exported regardless of the type of Entity { CDA, LCN, SCADA } being exported.

11

#### **Large Move Up: button**



Click this button to move the selected parameter item(s) up in the list by a page amount. The order in the list defines the column ordering in the generated Excel or CSV file. Press the key sequence Ctrl+Shift+Up as a shortcut to this button.



You can press the key sequence Ctrl+Shift+Home to move the selected parameter item(s) to the beginning of the list.

12

#### **Small Move Up: button**



Click this button to move the selected parameter item(s) up in the list one position. The order in the list defines the column ordering in the generated Excel or CSV file. Press the key sequence Ctrl+Up as a shortcut to this button.

13

## **Small Move Down: button**



Click this button to move the selected parameter item(s) down in the list one position. The order in the list defines the column ordering in the generated Excel or CSV file. Press the key sequence Ctrl+Down as a shortcut to this button.

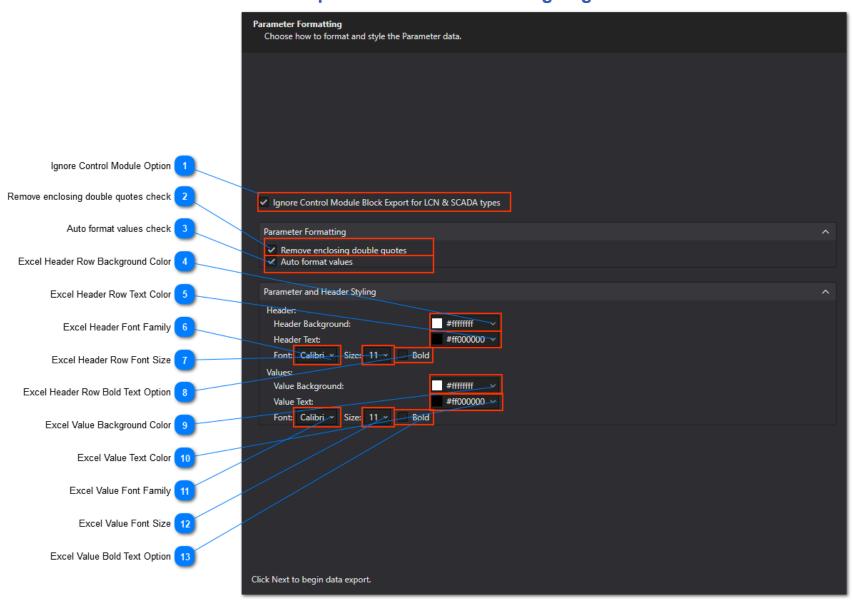
14

## **Large Move Down: button**



Click this button to move the selected parameter item(s) down in the list one position. The order in the list defines the column ordering in the generated Excel or CSV file. Press the key sequence Ctrl+Down as a shortcut to this button.

# **Data Export Parameter Formatting Page**



# Ignore Control Module Option

✓ Ignore Control Module Block Export for LCN & SCADA types

Choose this option to ignore the Control Module block when exporting LCN & SCADA type entities. This results in a single exported Data row for the entity. When importing LCN & SCADA type data the actual entity itself is embedded within a container "Control Module" to fit the same configuration schema as CDA type data. Users can still choose parameters that reside in this Control Module block, but they will now be merged into a single row of Excel data to represent the configuration entity.

# Remove enclosing double quotes check

Remove enclosing double quotes

Choose this option to remove double quote characters that enclose parameter values.

#### Auto format values check

✓ Auto format values

Choose this option to detect the value type and automatically format it based on its type.

## Excel Header Row Background Color



Click the drop-down button to choose the Excel Worksheet cell background color for the header row.

#### **Excel Header Row Text Color**



Click the drop-down button to choose the Excel Worksheet header row text color.

## Excel Header Font Family



Click the drop-down button to choose the font family to use for parameter values.



11 \*

Click the drop-down button to choose the font size for the header row text.

Excel Header Row Bold Text Option



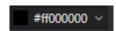
Choose this option to write all header row text as bold.

Excel Value Background Color



Click the drop-down button to choose the Excel Worksheet cell background color for parameter values.

Excel Value Text Color



Click the drop-down button to choose the Excel Worksheet parameter value text color.

Excel Value Font Family



Click the dropdown button to choose the font family to use for parameter values.

Excel Value Font Size



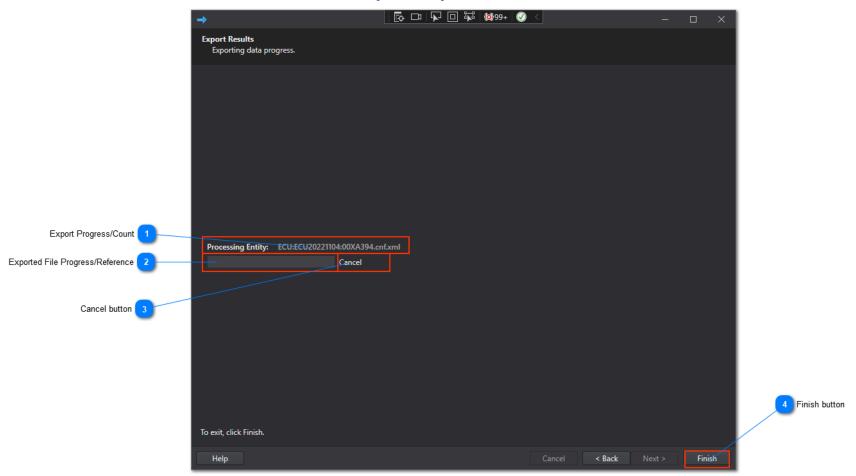
Click the drop-down button to choose the font size for parameter value text.

Excel Value Bold Text Option



Choose this option to write all parameter value text as bold.

# **Data Export Export Results**



Export Progress/Count

Processing Entity: ECU:ECU20221104:00XA394.cnf.xml

Displays the current entity being exported and when complete displays the count of Entity items that were written to the generated Excel file.

Exported File Progress/Reference

Current Progress indication.

Count of items exported: 589

C:\Temp\test.xlsx

Click this hyperlink file reference to open the generated Excel file.

Cancel button

Cancel

Click this button to cancel the data export operation.

Finish button

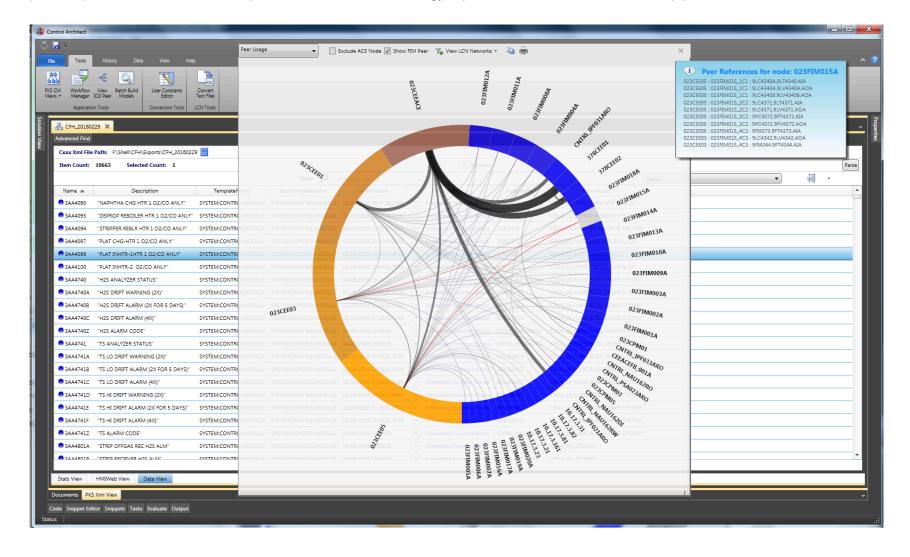
Finish

Click this button to close the Wizard view.

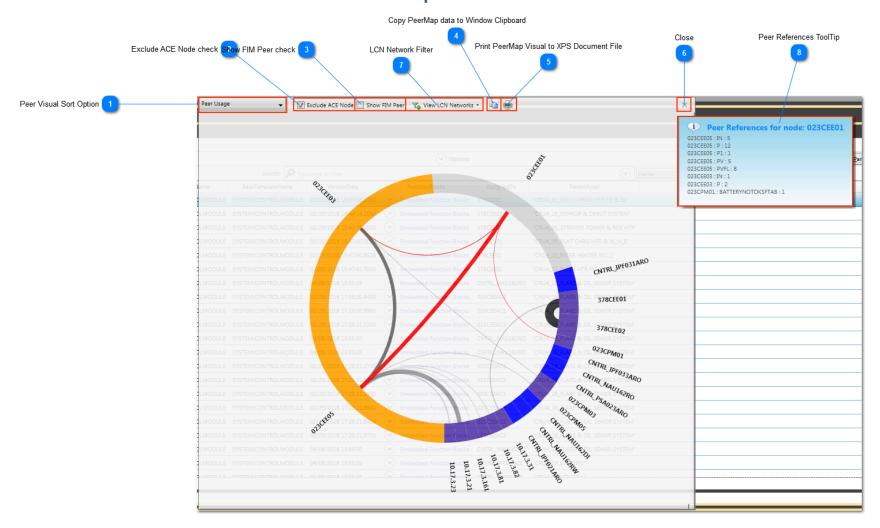
# **PeerMap**

The Peer Map shows a visual view of the peer connections within the Cxxx, ACE, PCDI, SCADA, and Fieldbus Interface Module (FIM) nodes within an EPKS Server.

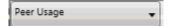
This view provides an indication of the amount of peer traffic that is occurring within the Server nodes. This view should be used to help identify potential peer traffic issues and to help validate future Control Strategy implementations with the intent to keep peer traffic to its minimum.



# **PeerMap Overview**



Peer Visual Sort Option



Use this option to change how the nodes are sorted by priority.

Exclude ACE Node check



Use this option to include/exclude peer connections to the ACE type nodes.

Show FIM Peer check



Use this option to include/exclude peer connection to FIM type nodes.

Copy PeerMap data to Window Clipboard



Click this button to copy the PeerMap data to the Window Clipboard for pasting into Excel. Use the Excel **Paste-Paste Special** function to copy the text object and image objects into your Excel Spreadsheet tabs.

When LCN node types exist in the Peer Map, only the selected LCN Networks and their relationships will be copied to the Clipboard.

Print PeerMap Visual to XPS Document File



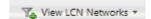
Click this button to print the visual contents of the PeerMap view to an XPS Document file.

Close



Click this button close the PeerMap view. You can reopen this view by again clicking the  $\stackrel{\text{def}}{=}$  button.

LCN Network Filter

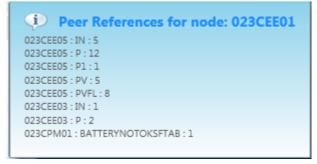


Click this drop-down menu button to select the LCN Networks to view. LCN Networks such as UCN's do not support peer communications between other UCN's.

During Analysis LCN Network types such as xPM's assigned to UCN's are identified for filtering. The first view of this Peer Map will default check a single LCN Network. You can choose to view one or more LCN networks using this filter.



#### **Peer References ToolTip**



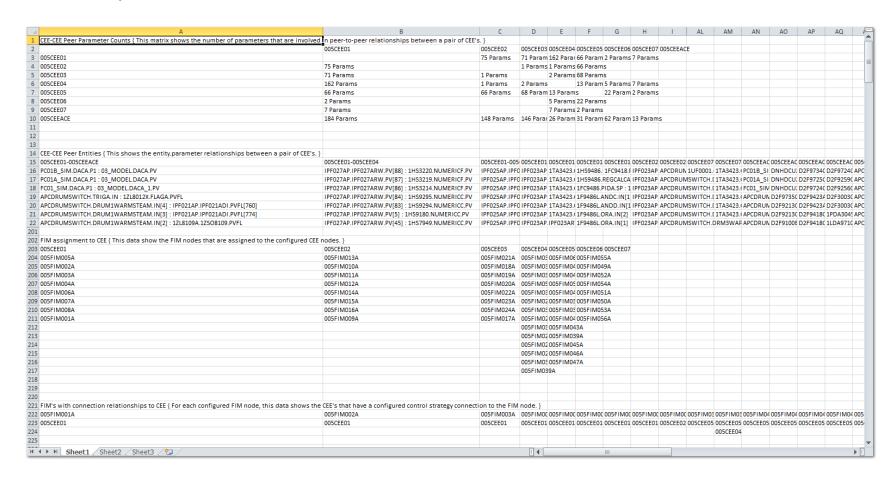
Hovering your mouse cursor over a node type in the Peer Map to show a ToolTip that indicates the Peer nodes and their Parameter types that are being referenced.

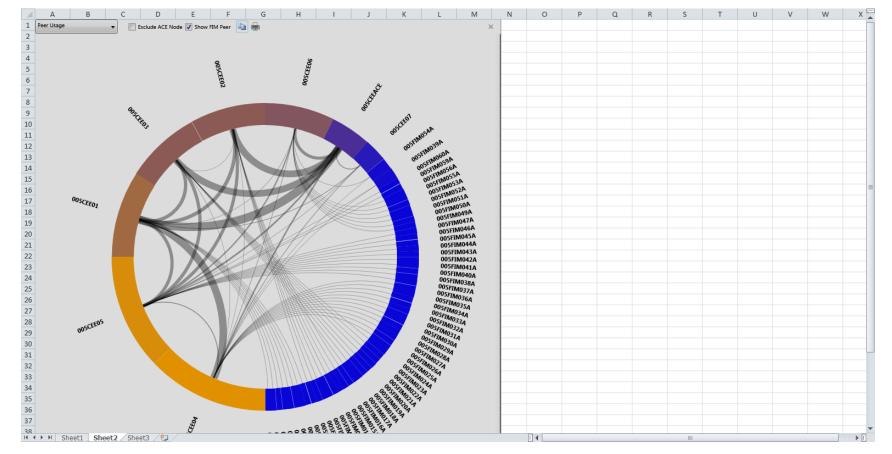
When you hover your mouse cursor over a node segment in the Peer Map, it will highlight all nodes that have peer relationships to it. The thickness or weight of the paths between nodes indicates the amount of Peer traffic.

# Peer Map Copy/Paste to Excel

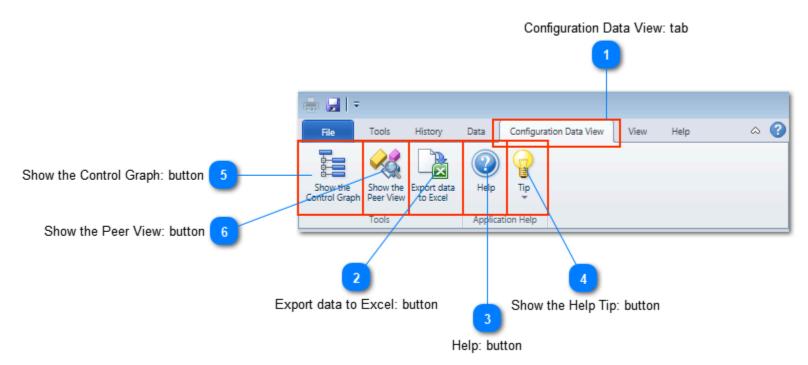
When copying Peer Map data to Excel, use the Excel **Paste-Paste Special** function to copy the text and bitmap image data from the Windows Clipboard into your Excel Workbook.

This example shows the text object copied from the Windows clipboard into an Excel Spreadsheet. Columns and rows have been hidden to show a summary view of the content.





#### **DataView Ribbon menu bar**



# Configuration Data View: tab



This Ribbon Toolbar tab contains the tool functions for the Data View. This Ribbon Toolbar is automatically displayed when the "Data View" tab of the active configuration view is focused.

# Export data to Excel: button



Click this button to export Control Module data to Excel using the Export Wizard.

### Help: button



Click this button to view the application Help topic for the "Data View".

### Show the Help Tip: button



Click this button to view quick help tips for the "Data View".

### Show the Control Graph: button



Click this button to toggle open/closed the Control Graph View for the selected Control Module items in the List View. The Control Graph will show relationships between Control Modules.

#### Show the Peer View: button



Click this button to toggle open/closed the Peer Graph View for the Configuration Asset Model (Project). This Peer View will show all Peer-to-Peer relationships between nodes that are contained in this Configuration Asset Model (Project).

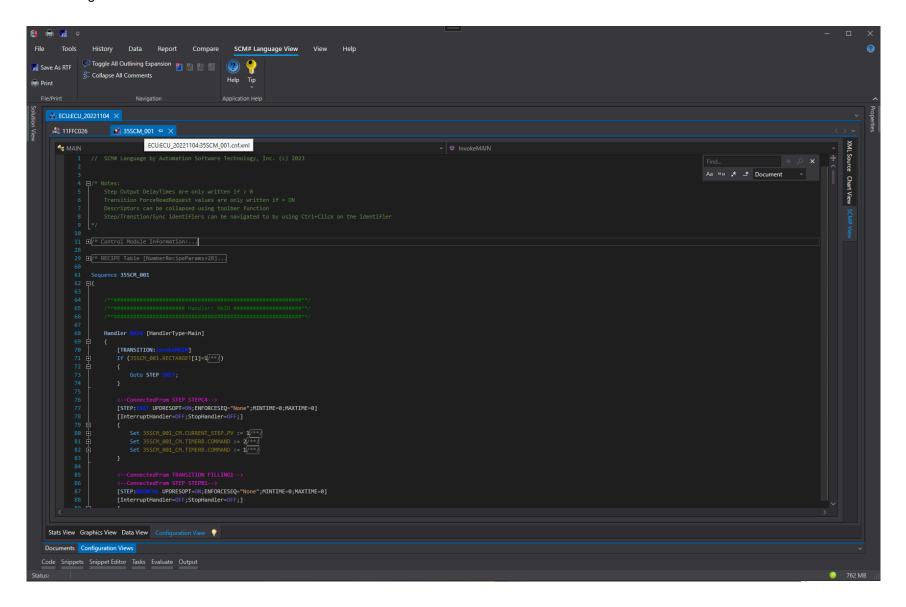
#### **SCM# View**

The SCM# View is an application XML configuration view for Cxxx SCM type Control Modules that provides an alternative view into the sequence configuration in a structured language similar to PMCL. The SCM# (pronounced as SCM Sharp) language is familiar to people with Honeywell Control Language experience, but offers a more structured approach in the use of delimiters such as curly braces to declare scope in code functions.

Working with large Sequence Control Modules can be difficult when navigating a large visual canvas of sequence elements or trying to discover entity and parameter usage within the module or simply following control flow branching. Features within the SCM# view facilitate search and navigation operations. You can also quickly identify changes between versions of a particular SCM entity using differential analysis tools on the textual SCM# source code.

The SCM# View is a read-only view of the SCM sequence configuration translated to SCM#. A future release of Control Architect will allow development of SCM modules using this structured SCM# Language.

You can access this view for an SCM type entity by invoking the ChartView. The SCM# view is automatically generated from analysis of the SCM configuration when the ChartView is invoked.



## Working in the SCM# View

Features within the SCM# view facilitate search and navigation operations between elements within the language view and ChartView. This view is currently a read-only type view and does not support editing. A future release of Control Architect will support the development of SCM type modules using SCM# language.

## **Navigation and Search Operations**

#### **Bookmarks**

You can add Bookmarks to the Document view to place a marker for a text line that can be be navigated using the Bookmark navigation buttons. Bookmark indicators are displayed in the Document view left margin indicator.

#### **Search and Highlight Operations**

- You can use the **Find** operation to highlight all occurrences of the entered text phrase. When the search text value changes or you click the **Find Next** button all found occurrences will be highlighted. The Document view right margin indicator will display glyphs that indicate their location in the document.
- You can select a label identifier such as a *STEP* or *TRANSITION* name and press the key sequence *Ctrl+F3* to highlight all occurrences of the selected label identifier in the document and auto-navigate to the next occurrence. The options in the **Find and Replace** Tool Window are used as the search options when using this feature.
- You can hold the Ctrl key while left mouse-clicking on an identifier to highlight all its occurrences within the document.

#### **Search Results Document Margin**

The Document view contains a right margin that is used to display search results. When a search operation is performed all highlighted occurrences of this item are displayed as glyphs in the margin indicator. The position in this indicator margin indicates its position within the source. You can use the vertical Scroll bar to scroll to a glyph position. You can also left mouse-click on a glyph to automatically scroll the view to its position. If you hover your mouse cursor on a glyph a ToolTip will be displayed that provides information related to the source line where the occurrence is found.

```
2KY3371_5 X
 🤻 MAIN
                                                 ■ InvokeMAIN
             SCM# Language by Automation Software Technology, Inc. (c) 2014
       2
       3
         4
      11
         12
      24
         25
      30
           Sequence 2KY3371_5
      31
      32
      33
      34
              35
              /**################### Handler: MAIN #################**/
      36
              37
              Handler MAIN [HandlerType=Main (/**/]]
      38
      39
                 [TRANSITION: InvokeMAIN /**/]
      40
      41
                 If (1)
      42
                 {
                     Then Goto STEP STEP1
      43
      44
      45
                 [STEP: STEP1/**/ UPDRESOPT=ON; ENFORCESEQ="None"; MINTIME=0; MAXTIME=0]
      46
      47
                 [InterruptHandler=OFF;StopHandler=OFF;]
      48
                 {
                     Set 2KY3371.UPDATE.PVFL := 0/**/
      49
      50
      51
                 [TRANSITION: TRANSITION2/**/]
      52
      53
                 If (2KY3371.DEVCTLA.GPV = 5/**/)
      54
                 {
                     Then Goto STEP STEP2
      55
      56
      57
      58
                 <--ConnectedFrom STEP STEP36-->
                 [STEP:STEP2/**/ UPDRESOPT=ON; ENFORCESEQ="None"; MINTIME=0; MAXTIME=0]
      59
                 [InterruptHandler=OFF;StopHandler=OFF;]
      60
                                                                      Line# 1116
                                                                      Set 2KY3371_S.RESADDRN[1] :=
                                                                      2KY3371_S.STEP1.NUM
             Chart View
      Data View
                     SCM# View
Source
Documents PKS Xml View
```

#### **Navigating Between Chart and SCM# Views**

You can navigate to the ChartView SCM sequence block from within the SCM# source document. You can also navigate from a ChartView sequence block to its SCM# code block.

From within the SCM# document view, place the editor caret within a code block and press the key sequence Ctrl+Shift while clicking within the code block with the left mouse button to navigate to its sequence block within the ChartView.

From within the ChartView, press the same key sequence *Ctrl+Shift* while clicking the sequence visual block with the left mouse button to navigate to its SCM# code block in the SCM# Document view.

#### **Printing and Saving the View**

To print the view, click the **Print** button.

To save the view as an RTF type document file, click the Save button. Saving to an RTF document will preserve all syntax coloring.

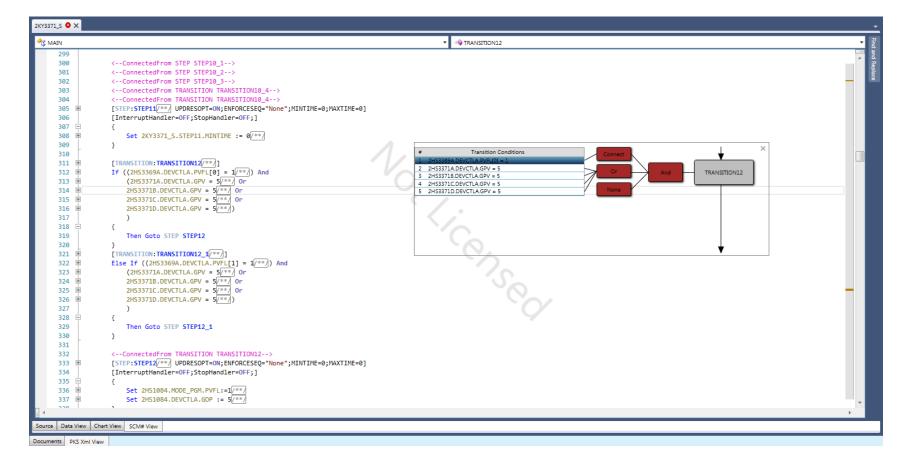
#### **Symbol Navigation Pane**

The editor provides a Symbol Navigation pane that tracks the current sequence Handler and sequence element within the Handler where the editor caret is placed. You can also quickly navigate to a Handler and a sequence element within this handler by selecting the Handler and Sequence Element from the drop-down lists in this pane.

### **View the Sequence Visual Block**

You can view the sequence visual block element that represents a particular code block such a TRANSITION or STEP by clicking within the code block to set the editor caret and pressing the key sequence *Ctrl+B*. You can move this Visual Block window by mouse clicking on an image surface in the window and dragging to a new location. If the Visual Block window is already visible and you select a new code block and press *Ctrl+B*, the Visual Block window will update its contents to reflect the current code block.

Copyright © 2024 by Automation Software Technology, Inc. All Rights Reserved.

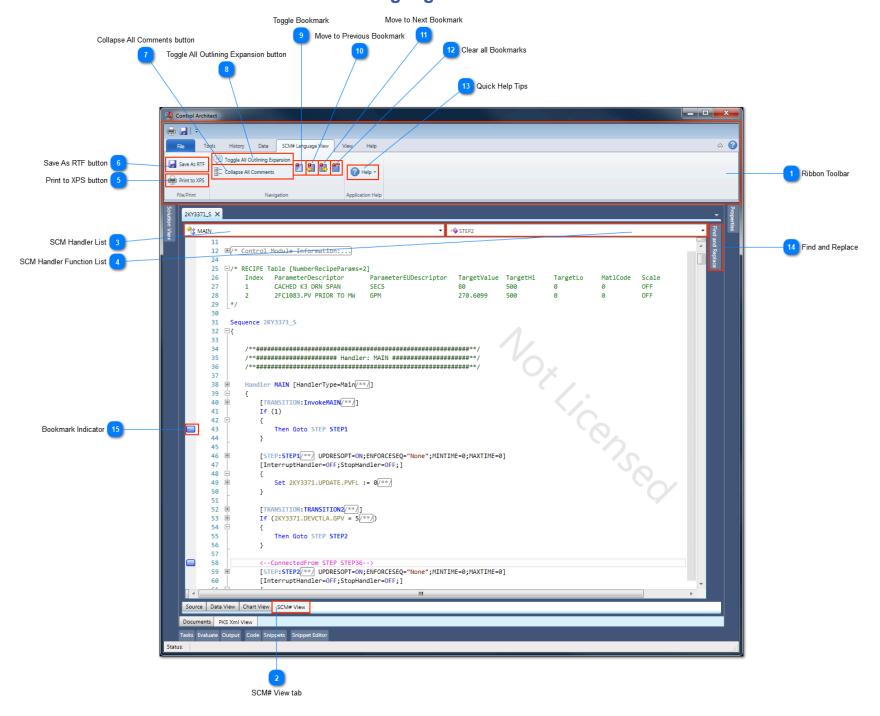


## **Source Code Comments Operations**

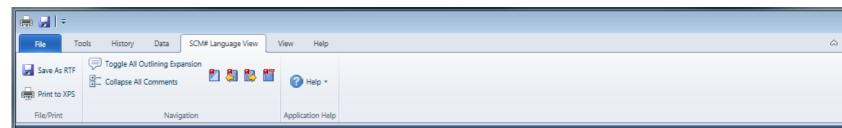
Comments found in the sequence block configurations are translated to SCM# as code comments. These comments are enclosed within comment blocks as in  $/*This\ is\ a\ comment*/$ . Comment blocks form a comment region that can be expanded and collapsed. A region button is placed to the left for each source line that contains a comment. All comments are collapsed by default when viewing the SCM# code document. A collapsed code comment is indicated with the shortcut text /\*\*/. A comment can be quickly expanded by a left mouse double-click on the comment shortcut text. To collapse the expanded text comment toggle the region button for the code line.

You can click the Toolbar button **Toggle All Outlining Expansion** to expand all code comments. You can also click the the Toolbar button **Collapse All Comments** to collapse all comments in the source document.

# **SCM# Language View Overview**



Ribbon Toolbar



SCM# View tab

SCM Handler List

**Ŷ**\$ MAIN ▼

Drop-down list to quickly navigate to an SCM handler definition. The handler item displayed here tracks the current cursor position in the SCM document view.

SCM Handler Function List

Drop-down list to quickly navigate to an SCM handler function definition {STEP, TRANSITION, SYNC}. The handler item displayed here tracks the current cursor position in the SCM document view.

Print to XPS button

Click this button to print the SCM Document view to an XPS Document file.

Save As RTF button



Click this button to save the SCM Document view to an RTF formatted Document file. This preserves all syntax coloring.

Collapse All Comments button



Click this button to collapse all document comments to their /\*\*/ collapsed view.

Toggle All Outlining Expansion button



Click this button to expand/collapse all document regions including text comments.

Toggle Bookmark



Click this button to toggle a Bookmark for the current active line in the Document view.

Move to Previous Bookmark



Click this button to scroll the Document view to the previous Bookmark.

Move to Next Bookmark



Click this button to scroll the Document view to the next Bookmark.

Clear all Bookmarks



Click this button to remove all Bookmarks from the Document view.

Quick Help Tips



Click this button to view quick help tips for working with the Document view.

Find and Replace

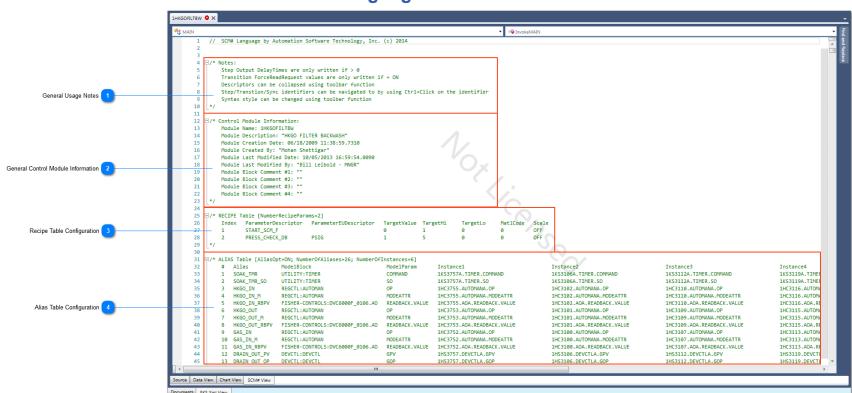


Click on this tab to expand the Tool Window that contains the text "Find" functions.

Bookmark Indicator



# **SCM# Language Source Overview**



# General Usage Notes

# General Control Module Information

```
□/* Control Module Information:

Module Name: 1HKGOFILTBW

Module Description: "HKGO FILTER BACKWASH"

Module Creation Date: 06/18/2009 11:38:59.7310

Module Created By: "Mohan Shettigar"

Module Last Modified Date: 10/05/2013 16:59:54.0090

Module Last Modified By: "Bill Leibold - MNGR"

Module Block Comment #1: ""

Module Block Comment #2: ""

Module Block Comment #3: ""

Module Block Comment #4: ""
```

# Recipe Table Configuration

```
□/* RECIPE Table [NumberRecipeParams=2]

Index ParameterDescriptor ParameterEUDescriptor TargetValue TargetHi TargetLo MatlCode Scale

1 START_SCM_F 0 1 0 0 0FF

2 PRESS_CHECK_DB PSIG 1 5 0 0 0FF

*/
```

# Alias Table Configuration

/* A	LIA	S Table [AliasOp	ot=ON; NumberOfAliases=26; NumberOf	[nstances=6]		
	#	Alias	ModelBlock	ModelParam	Instance1	Instance2
	1	SOAK_TMR	UTILITY:TIMER	COMMAND	1KS3757A.TIMER.COMMAND	1KS3106A.TIMER.COMMAND
	2	SOAK_TMR_SO	UTILITY:TIMER	SO	1KS3757A.TIMER.SO	1KS3106A.TIMER.SO
	3	HKGO_IN	REGCTL: AUTOMAN	OP	1HC3755.AUTOMANA.OP	1HC3102.AUTOMANA.OP
	4	HKGO_IN_M	REGCTL: AUTOMAN	MODEATTR	1HC3755.AUTOMANA.MODEATTR	1HC3102.AUTOMANA.MODEATTR
	5	HKGO_IN_RBPV	FISHER-CONTROLS:DVC6000F_0106.AO	READBACK.VALUE	1HC3755.AOA.READBACK.VALUE	1HC3102.AOA.READBACK.VALUE
	6	HKGO_OUT	REGCTL: AUTOMAN	OP	1HC3753.AUTOMANA.OP	1HC3101.AUTOMANA.OP
	7	HKGO_OUT_M	REGCTL: AUTOMAN	MODEATTR	1HC3753.AUTOMANA.MODEATTR	1HC3101.AUTOMANA.MODEATTR
	8	HKGO_OUT_RBPV	FISHER-CONTROLS:DVC6000F_0106.AO	READBACK.VALUE	1HC3753.AOA.READBACK.VALUE	1HC3101.AOA.READBACK.VALUE
	9	GAS_IN	REGCTL: AUTOMAN	OP	1HC3752.AUTOMANA.OP	1HC3100.AUTOMANA.OP
	10	GAS_IN_M	REGCTL: AUTOMAN	MODEATTR	1HC3752.AUTOMANA.MODEATTR	1HC3100.AUTOMANA.MODEATTR
	11	GAS_IN_RBPV	FISHER-CONTROLS:DVC6000F_0106.AO	READBACK.VALUE	1HC3752.AOA.READBACK.VALUE	1HC3100.AOA.READBACK.VALUE
	12	DRAIN_OUT_PV	DEVCTL:DEVCTL	GPV	1HS3757.DEVCTLA.GPV	1HS3106.DEVCTLA.GPV
	13	DRAIN_OUT_OP	DEVCTL:DEVCTL	GOP	1HS3757.DEVCTLA.GOP	1HS3106.DEVCTLA.GOP

Instance

1KS3119A

1HC3116.

1HC3116.

1HC3115.

1HC3115.

1HC3115.

1HC3113.

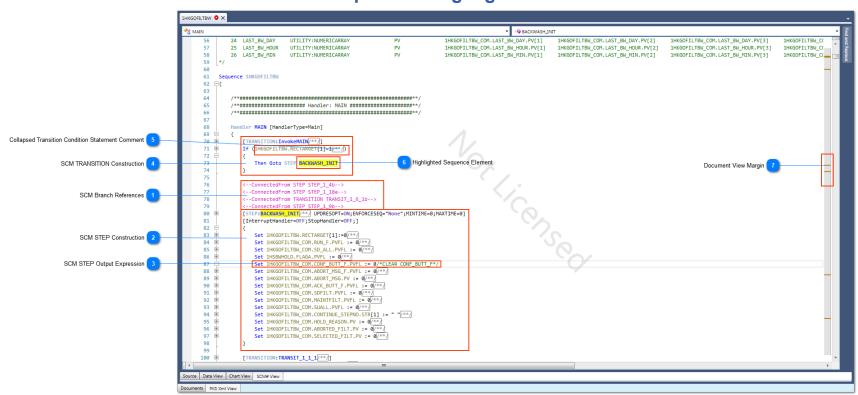
1HC3113.

1HC3113.

1HS3119.

1HS3119.

# **SCM# Sequence Language Overview**



# SCM Branch References

```
<--ConnectedFrom STEP STEP_1_4b-->
<--ConnectedFrom STEP STEP_1_10a-->
<--ConnectedFrom TRANSITION TRANSIT_1_6_1b-->
<--ConnectedFrom STEP STEP 1 9b-->
```

These source statements indicate where SCM Steps and Transitions branch to this STEP: BACKWASH\_INIT.

## SCM STEP Construction

```
[STEP: BACKWASH_INIT /** / UPDRESOPT=ON; ENFORCESEQ="None"; MINTIME=0; MAXTIME=0]
[InterruptHandler=OFF;StopHandler=OFF;]
              Set 1HKGOFILTBW.RECTARGET[1]:=0/**/
             Set 1HKGOFILTBW_COM.RUN_F.PVFL := 0/**/
             Set 1HKGOFILTBW_COM.SD_ALL.PVFL := 0/**/
             Set 1HSBWHOLD.FLAGA.PVFL := 0/**/
              Set 1HKGOFILTBW_COM.CONF_BUTT_F.PVFL := 0/*CLEAR CONF_BUTT_F*/
              Set 1HKGOFILTBW_COM.ABORT_MSG_F.PVFL := 0 (**/
             Set 1HKGOFILTBW_COM.ABORT_MSG.PV := 0(**/
             Set 1HKGOFILTBW_COM.ACK_BUTT_F.PVFL := 0(/**/
             Set 1HKGOFILTBW_COM.SDFILT.PVFL := 0/**/
             Set 1HKGOFILTBW_COM.MAINTFILT.PVFL := 0/**/
             Set 1HKGOFILTBW_COM.SUALL.PVFL := 0 (**/
              Set 1HKGOFILTBW_COM.CONTINUE_STEPNO.STR[1] := " "[/**/
              Set 1HKGOFILTBW_COM.HOLD_REASON.PV := 0/**/
              Set 1HKGOFILTBW_COM.ABORTED_FILT.PV := 0 (/**/
              Set 1HKGOFILTBW_COM.SELECTED_FILT.PV := 0 \( \text{ \
}
```

This STEP construction shows how an SCM STEP function is converted to its source view.

# SCM STEP Output Expression

```
Set 1HKGOFILTBW_COM.CONF_BUTT_F.PVFL := 0/*CLEAR CONF_BUTT_F*/
```

This is a typical SCM Step Output expression with expanded comment.

#### SCM TRANSITION Construction

```
[TRANSITION:InvokeMAIN[/**/]
If (1HKGOFILTBW.RECTARGET[1]=1[/**/])
{
    Then Goto STEP BACKWASH_INIT
}
```

Transition functions are translated using If/Then/Elself code statements.

## Collapsed Transition Condition Statement Comment

```
1HKGOFILTBW.RECTARGET[1]=1/**/)
```

Comments by default are collapsed. Hovering the mouse cursor over a collapsed comment will show its contents as a ToolTip. You can also click the region + expand button for the line comment to show it. Left mouse double-click on a comment will also expand the comment region to show the comment.

Copyright © 2024 by Automation Software Technology, Inc. All Rights Reserved.



# **Highlighted Sequence Element**

## BACKWASH\_INIT

When searching the document for text, search results will be highlighted to show their position within the document. The document view margin located to the right of the document view will indicate where all occurrences of this search text exists. You can also left mouse-click while holding the Ctrl key down to quickly find and highlight all occurrences of an SCM sequence element Id.

7

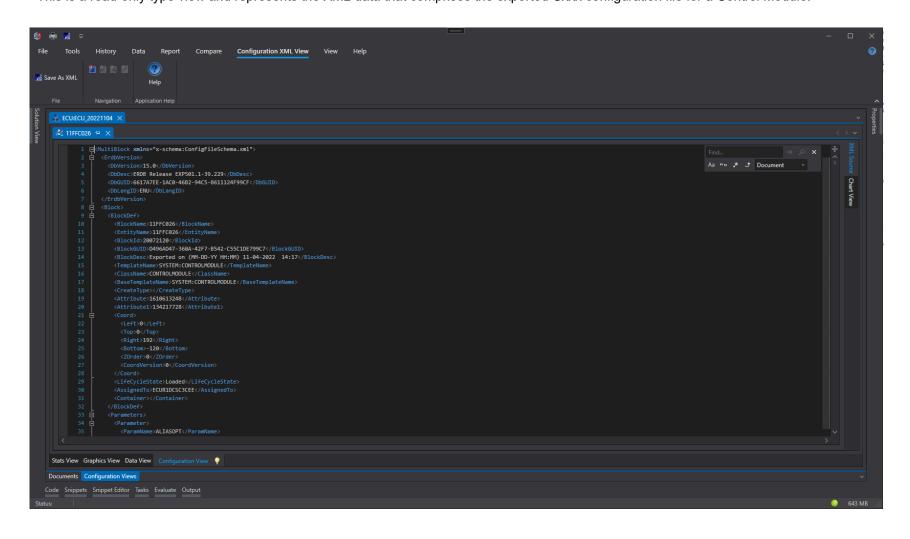
## **Document View Margin**



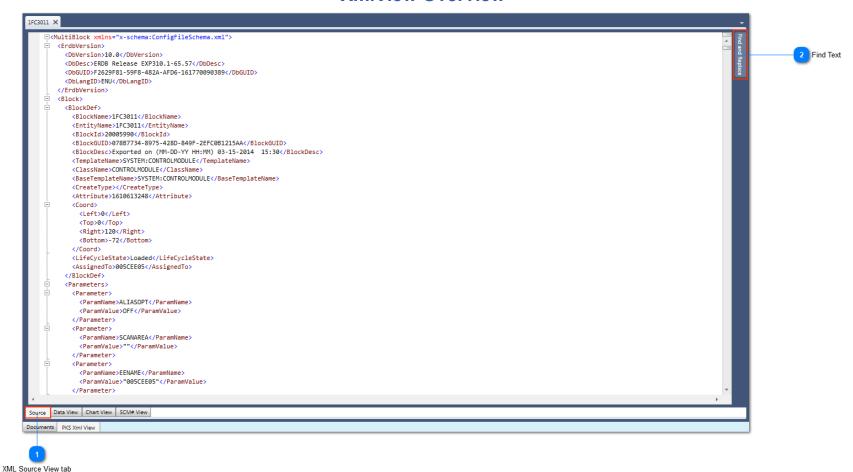
Indicates where searched text references exist within the document view. You can hover mouse cursor over the highlight glyph to view a ToolTip that shows the source line contents. You can also left mouse-click on a glyph to navigate to the view source.

#### **XmlView**

The XML configuration view displays the raw XML source code for the Control Module entity item that is currently being viewed in the ChartView. This is a read-only type view and represents the XML data that comprises the exported Cxxx configuration file for a Control Module.



### **XmlView Overview**



1

## **XML Source View tab**

Source

2

## **Find Text**



Click or hover your mouse cursor over this tab to show the Find Text feature. Searched text values can be navigated in the XML source view using F3 to navigate forward and Shift+F3 to navigate backwards in the XML document.

# **Configuration Asset Model Generation**

Control Architect requires that Cxxx XML configuration files be parsed to an internal Configuration Asset data Model. This data model is used throughout Control Architect to present data views and support various tools and functions. To begin using PKS configuration types in Control Architect you must first import the configuration data into the <u>document Server</u>. The Import function supports both folder path and archive file as the source of the exported configuration files.

Control Architect also supports converting LCN Honeywell Exception Build (EB) & Graphic DX files, PKS Server Scada exported PNT & HDW file types and Honeywell FSC Safety Controller configuration Excel files. The Text File conversion tool supports reading these point type files and converting their entities to PKS XML compliant configuration files that are stored on the document Server.

## **Generating Configuration Asset Models**

Configuration Asset models can be generated using the following functions:

- 1. PKS Server exported Cxxx XML configuration using the Import function.
- 2. PKS Server Scada exported PNT & HDW file types and LCN exported file types using the Text File Conversion tool.

# **Parsing Failure**

## **Causes**

Parsing failures will almost always be due to a corrupted Xml format.

## Resolution

The only resolution is to re-export the corrupted entity from the PKS Server and try to re-import the XML project.

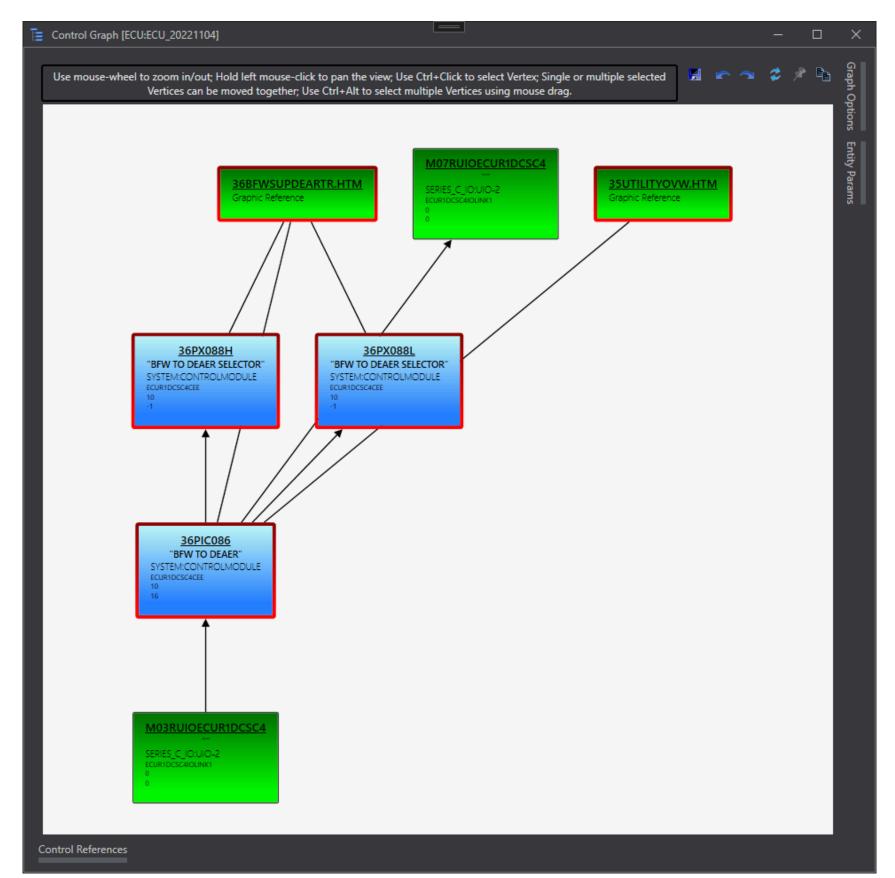
## **PKS Control Graph**

#### **Overview**

The Control Graph provides a visual rendering of Control Modules that contain parameter connection relationships that form a Control Strategy. The Control Graph view analyzes selected Control Module entity connections to form a visual block rendering of Control Module entities with all their parameter connection relationships rendered as data flow connections.

As Control Strategy development becomes more granular using principles of object oriented development often cohesive functions are grouped within a Control Module such as I/O processing, logical functions, regulatory PID control, regulatory Override control, etc. that results in a Control Strategy comprised of multiple Control Modules. Control System vendors typically provide a Control Strategy development view that supports the development of a single Control Module where parameter connection references are supported to provide data flow connections external to the Control Module. The PKS Experion *Control Builder* application also supports hierarchical development where a single Control Module can act as a container to multiple Control Modules acting as embedded CM's providing a cohesive view of a Control Strategy. As with any good control development there is a cost to doing it better or more correct that often is a hard sell in this economy and competition amongst 3rd party integrators.

The *PKS Control Graph* view provides a summary view of Control Modules that have connection relationships that form a Control Strategy. It provides a high level overview of your control system applications allowing you to quickly visualize relationships and anomalies.



#### **Block Color Representation**

Block colors are used to provide visual identification to the function of the block within the Control Graph.

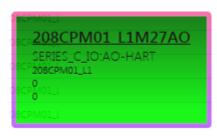
This block color gradient is used to indicate a block that resides in a non-ACE CEE and its magenta border also indicates that it is the source of the connection relationship analysis.

2AA4873 SYSTEM:CONTROLMODULE 208CEE03 362 -1

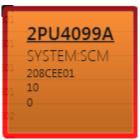
This block color gradient is used to indicate a block that resides in an ACE CEE and its red border color also indicates that its CEE is not the same as the source of the Control Module that was used to display the Control Graph.

2AC4086X SYSTEM:CONTROLMODULE ACECEE\_SIM 1470 -1

This block color gradient is used to indicate a block that is an I/O type or a Process graphic reference.



This block color gradient is used to indicate an SCM type Control Module and its red border color also indicates that its CEE is not the same as the source of the Control Module that was used to display the Control Graph.



# **Using the PKS Control Graph**

## **Using the Control Graph**

The Control Graph provides various visual cues that help to quickly identify potential anomalies and data relationships between Control Strategies that are difficult to find using other analysis methods. The following graphical images are used to identify how these visual cues are presented when viewing Configuration Asset Model data within *Control Architect*.

#### **Displaying a Control Graph**

The **PKS CM Views** function provides several view types for viewing PKS Cxxx Control Strategy configuration. The Data View tab provides a listing of all entities exported from an Experion Server that have been parsed into a Configuration Asset data Model. To view a Control Graph select an entity(s) from the list and left mouse-click the button in the top right area of the tab view to analyze the data relationships for the selected Control Module items and present a Control Graph of the analysis results.

Hint: Hover the mouse over button targets to view their ToolTips that provide a description of their actions.

#### **Visualizing Control Strategies**

You can keep the Control Graph view open while scrolling through the list of entities listed in the DataView tab. Each time a new selection is detected, the Control Graph view will be automatically updated with the current connection relationship analysis.

You can also multiple select entities from the list in the DataView tab and view their combined connection relationships analysis.

#### **Changing the Layout**

Use mouse-wheel to zoom in/out and hold the left mouse-click and drag to pan the view.

Vertices (Blocks) can be dragged and moved to adjust their position in the graph along with their attached edges (connections).

You can keep pressed the Alt key while dragging with the left mouse cursor to select a group of blocks to zoom to and fill the view.

You can change the opacity of the Control Graph view background in the Application options.

#### **Control Graph References**

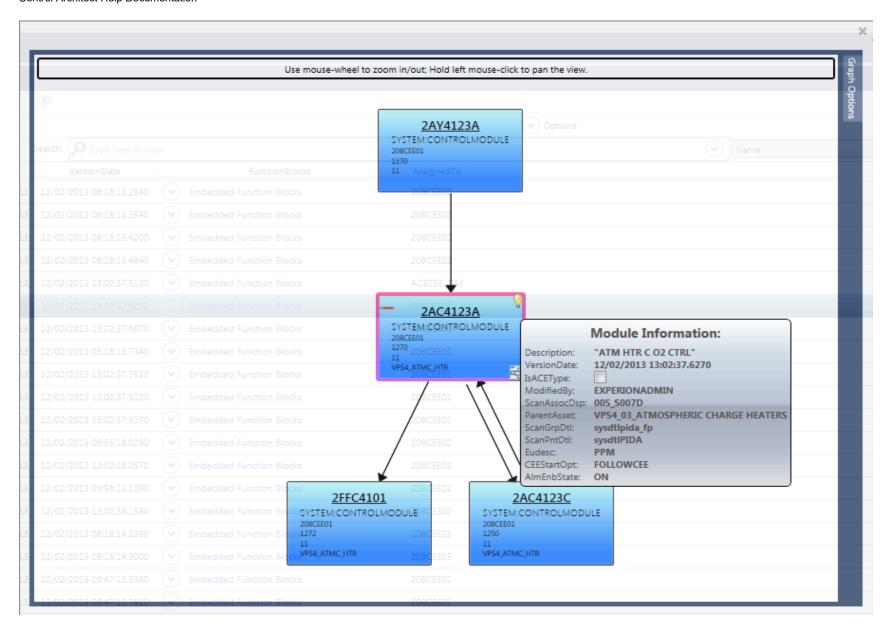
Control Graph references can be viewed in the Control References tab.

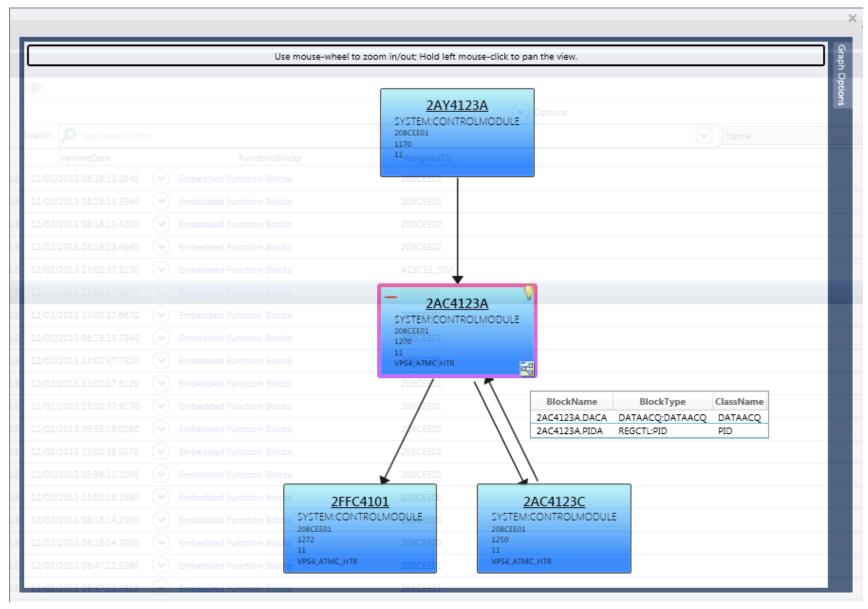
### Navigating to a Block ChartView

You can left mouse button double-click on a block to navigate to the ChartView for the clicked Control Module.

#### Working with the Block Adorner

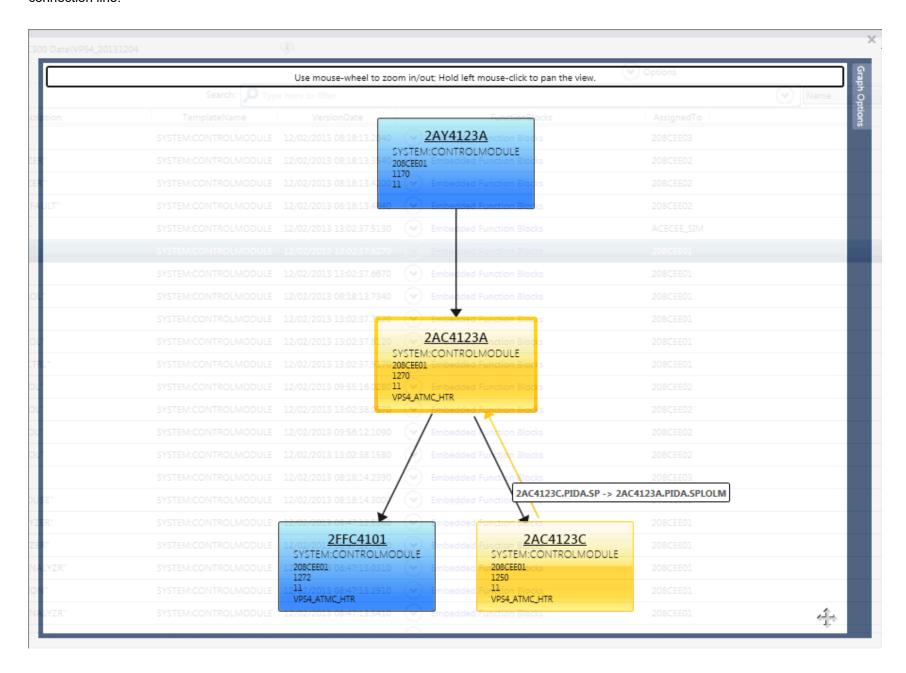
The following image show an adorner that is presented when the mouse cursor is hovered over a block that represents a Control Module in the Control Graph. The adorner provides visual cues such as image targets that when hovered over with the mouse will present more data associated with the particular Control Module block such as common parameter data values, list of external connection references, and a list of embedded function blocks. The adorner also provides an expand/collapse button that allows you to delve deeper into the connection relationships for the particular module. The magenta border around the Control Module block 2AC4123A indicates that this was the source of the analysis and it is currently expanded showing all immediate connection relationships to itself.



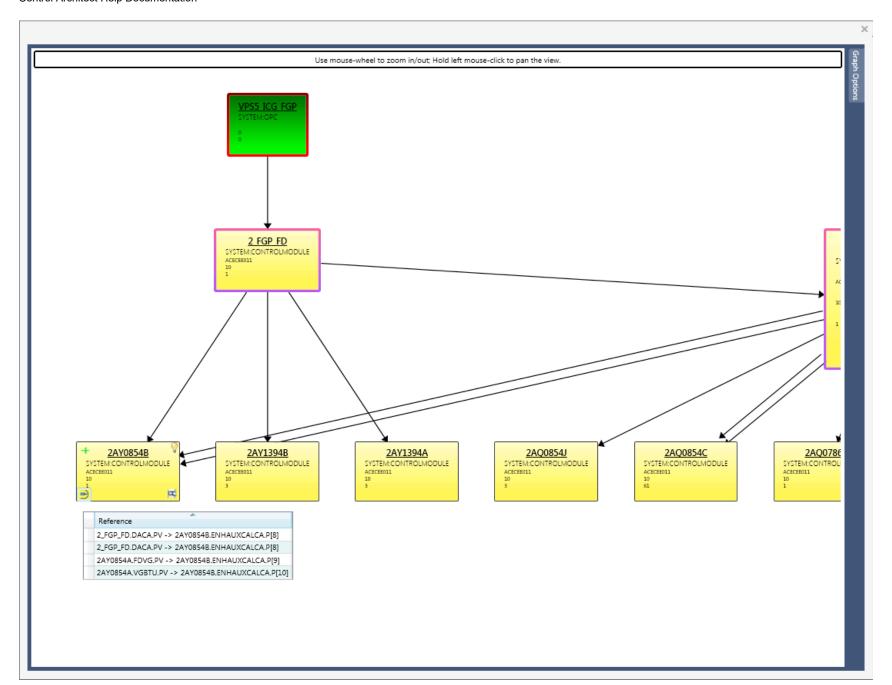


Blocks relationships can be further expanded and collapsed by clicking the button located in the top left corner of the block. When blocks are expanded to show their immediate connection relationships the border color of the block will change to the magenta color to indicate that it is a source of connection relationship analysis.

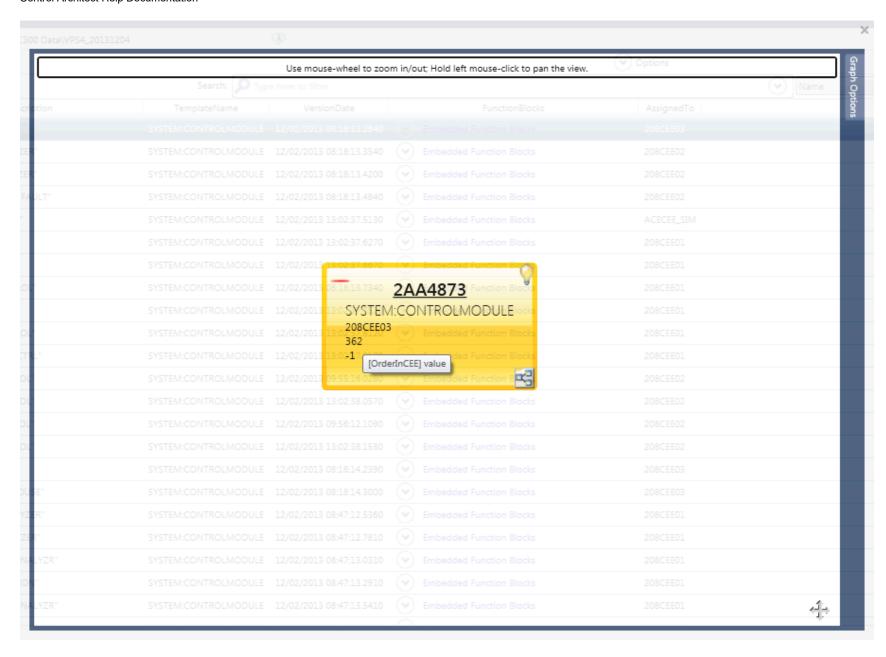
The image below shows a connection ToolTip that is displayed when a connection edge is hovered over with the mouse. When a connection edge is hovered over using the mouse cursor, the blocks that are the source and target of the connection will become highlighted along with the connection line.



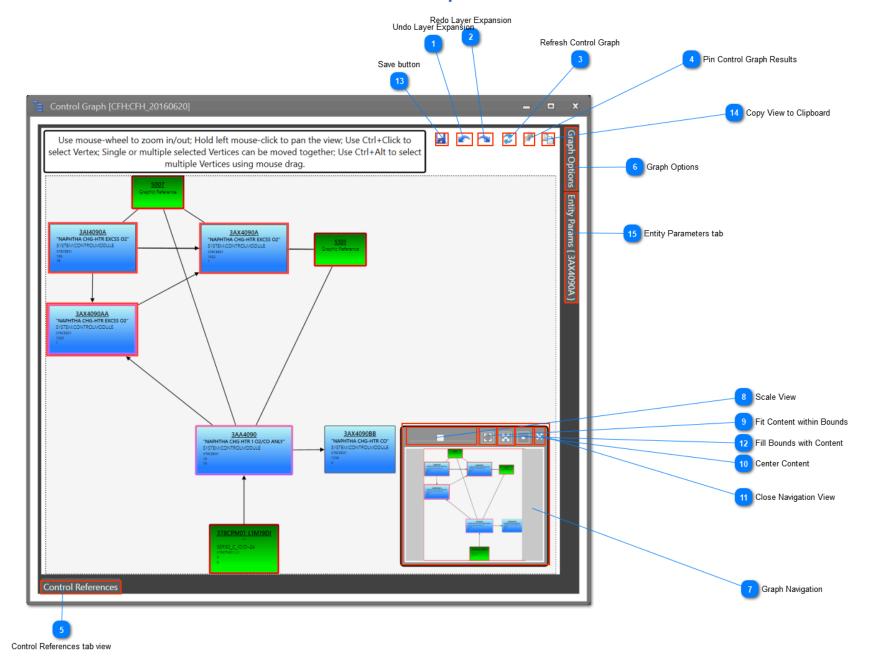
The image below shows a ToolTip that displays a list of external connection references when the mouse is hovered over the bottom left adorner image. Notice the image in the top left location of the adorner that indicates that this block can be delved (expanded) into to show its immediate external connected Control Module blocks.



The image below shows a ToolTip that is displayed when the mouse cursor is hovered over the value 362, which indicates that this value represents the OrderInCEE parameter. Hovering the mouse cursor over the item values displayed in a block will provide a brief description of the value or in the case of the Control Module name its ToolTip will display the Description parameter value for the Control Module.



## **Control Graph View**



# Undo Layer Expansion



Click this button to undo the last layer expansion. Ctrl+Z key combination is also supported.

# Redo Layer Expansion



Click this button to redo the last layer expansion that was undone using the Undo function.. Ctrl+Y key combination is also supported.

# Refresh Control Graph



Click this button to regenerate the Control Graph results. If you have performed a number of expansion events to further delve into a Control Strategy, refreshing the graph will force a regeneration of the Control Graph contents based on the current selected item(s) in the DataView.

## Pin Control Graph Results

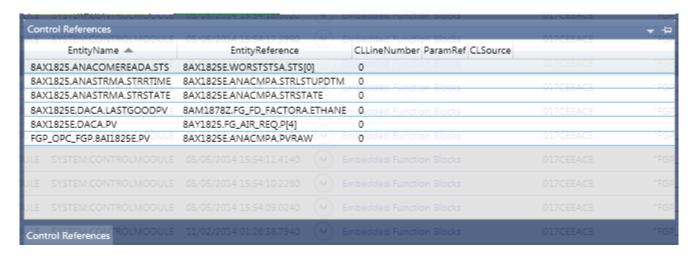


Toggle this to pin/unpin the freeze status of the current Control Graph generated results. Pinning the graph will allow you to navigate the DataView without having the Control Graph auto-update to reflect the selected item change in the DataView.

## Control References tab view

# Control References

Displays a list of all Control relationships that were detected while generating the current Control Graph. This is a auto-hide tab view that can be docked.

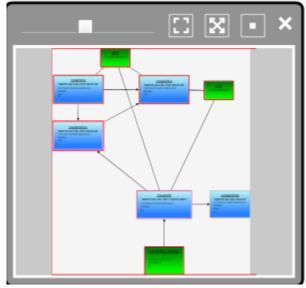


# Graph Options



Provides a set of options that allow you to define various behaviors associated with the Control Graph such as how connections are displayed, algorithms used to generate the Vertex node layout, edge routing algorithms that define how edges are routed, animation, default scaling, Vertex layer gap spacing, etc. See the topic <u>Control Graph Options</u>.

# Graph Navigation



Provides navigation functions that allow you to scale, pan, fill content to bounds, and center content.

### Scale View



Drag the Slider to scale the contents of the graph view.

# Fit Content within Bounds



Click this button to fill the current graph view with its contents assuring that all Vertex nodes are displayed within the bounds of the view.

### Center Content



Click this button to center the content (vertex nodes) of the graph within the view.

## Close Navigation View



Click this button to close the Navigation View.



## **Fill Bounds with Content**



Click this button to fill the current graph view with its contents.

13

## Save button



Click this button to <u>save</u> the Control Graph View state to the configuration project. Saved Control Graph view states are automatically restored when a Control Graph view is requested for a single Entity item in the Data View list.

14

# **Copy View to Clipboard**



Click this button to copy the Control Graph View as an image for pasting into applications such as Word.

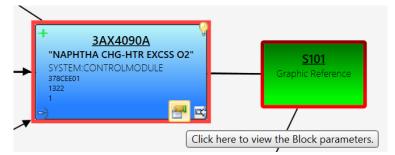
15

## **Entity Parameters tab**

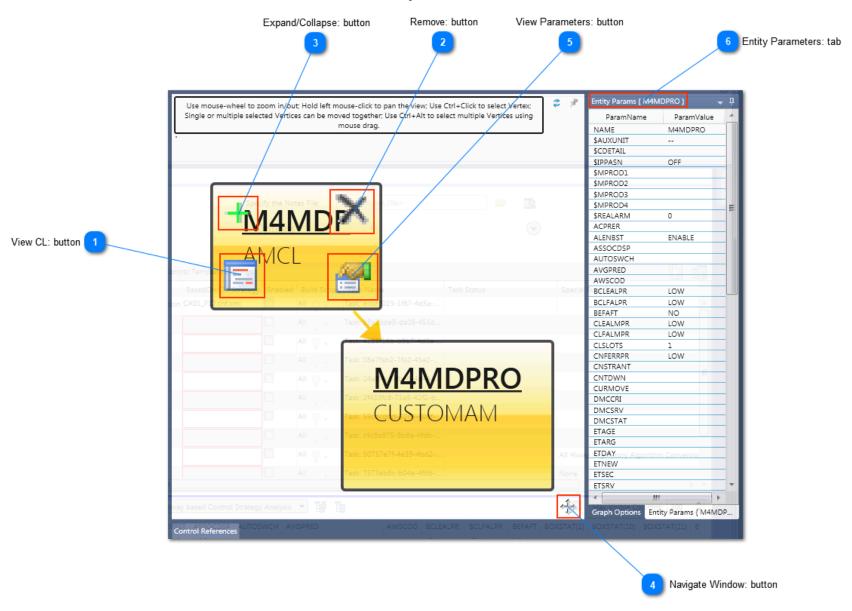


This Tab becomes visible when a request is made to view the configuration Parameters for a Block in the Control Graph view.

The highlighted button on the block adorner below showing its ToolTip is clicked to view its configuration Parameters.



## **Control Graph Vertex Block**



View CL: button



Click this button to view the CL file associated with this Entity Vertex Block. AM type entities may have either AMCL or Package files that are displayed in a CL View window when clicked. CL Package files are displayed in a list that can be selected for viewing its file.

Remove: button



Click this button to remove this Vertex Block from the Control Graph surface. The original contents of the graph can be restored by clicking the Refresh button.

Expand/Collapse: button



Click this button to expand or collapse entity relationships to this Vertex Block (Entity). When Entity Vertex blocks are expanded or collapsed, the Graph Control will redraw using the current layout algorithm to show the change in Entity relationships.

Navigate Window: button



Click this button to restore the Navigation View.

View Parameters: button



Click this adorner button to view the parameters for the Entity represented by this Vertex Block.

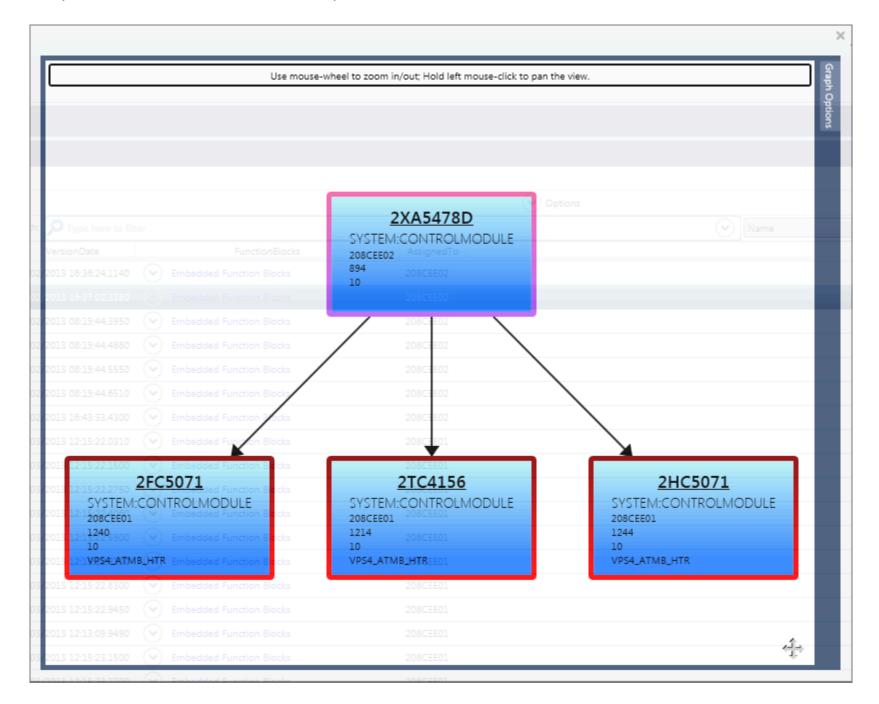


Click or hover over this tab with your mouse cursor to view the Entity parameters.

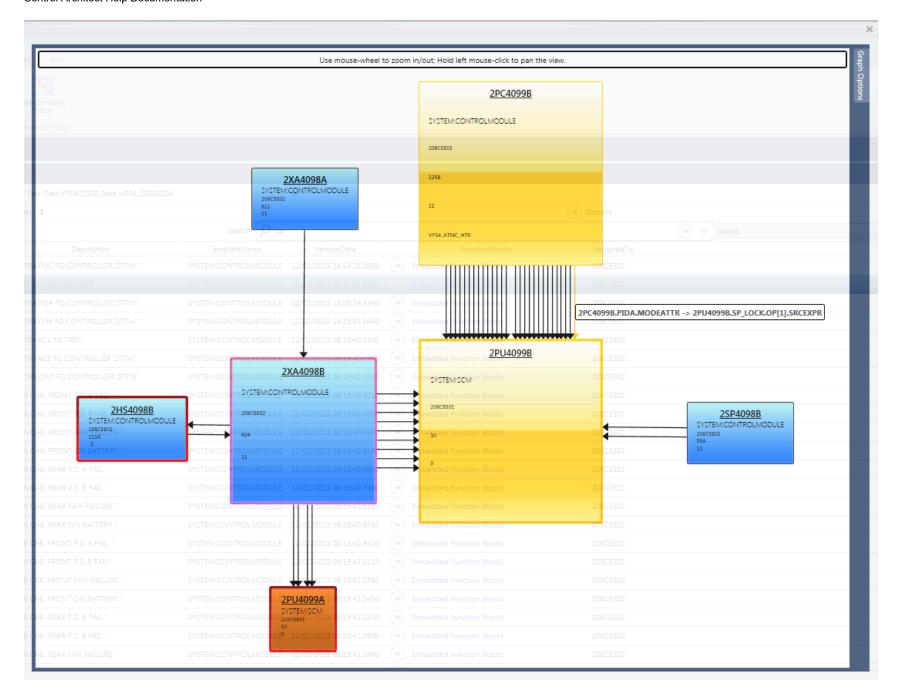
# **PKS Control Graph View Examples**

## **Examples of Using the Control Graph to Find Anomalies**

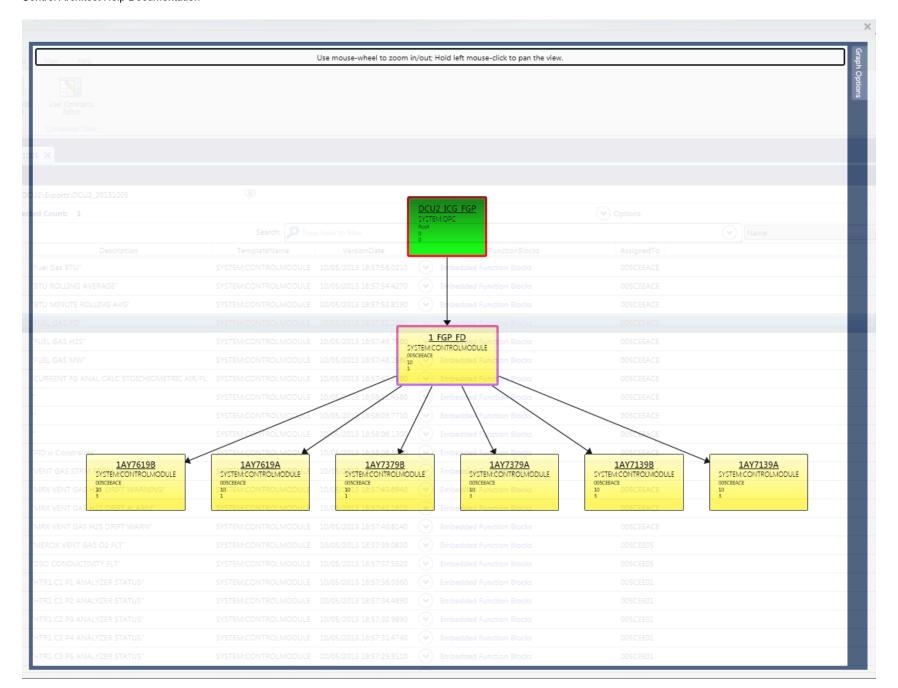
Here is an example of where CEE-CEE peer could have been reduced by moving the Control Module 2XA5478D to the CEE 208CEE01. The Control Graph image below indicates that the Control Modules 2FC5071, 2TC4156 and 2HC5071 reside in another CEE other than the source of the analysis Control Module 2XA5478D as indicated by their red border color.



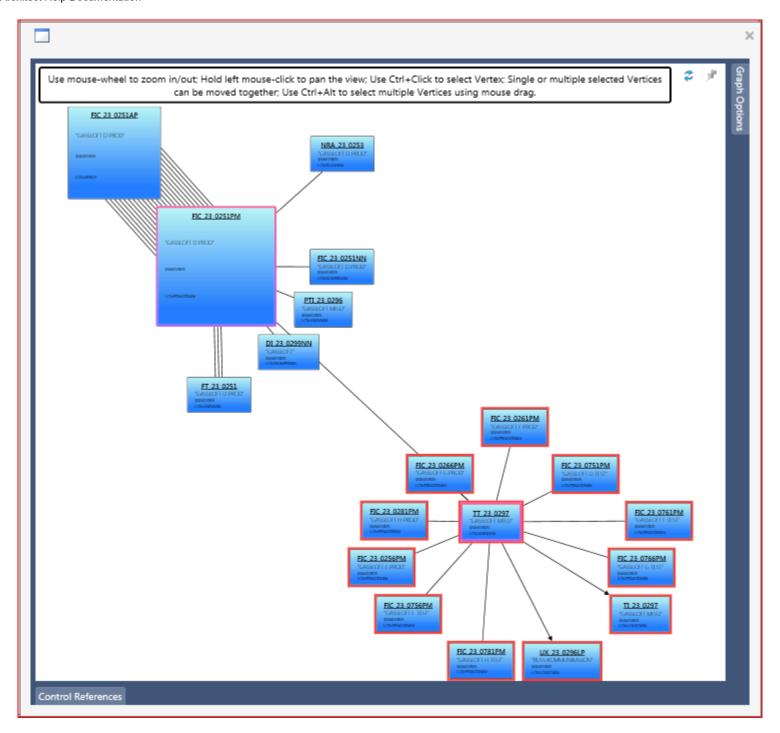
Here is an example of a Control Strategy view that contains modules including SCM's that have Peer-to-Peer connections that could have been eliminated with proper control design and I/O partitioning.



Here is an example of an ACE resident Control Strategy referencing parameter data through an Intercluster Gateway (ICG).

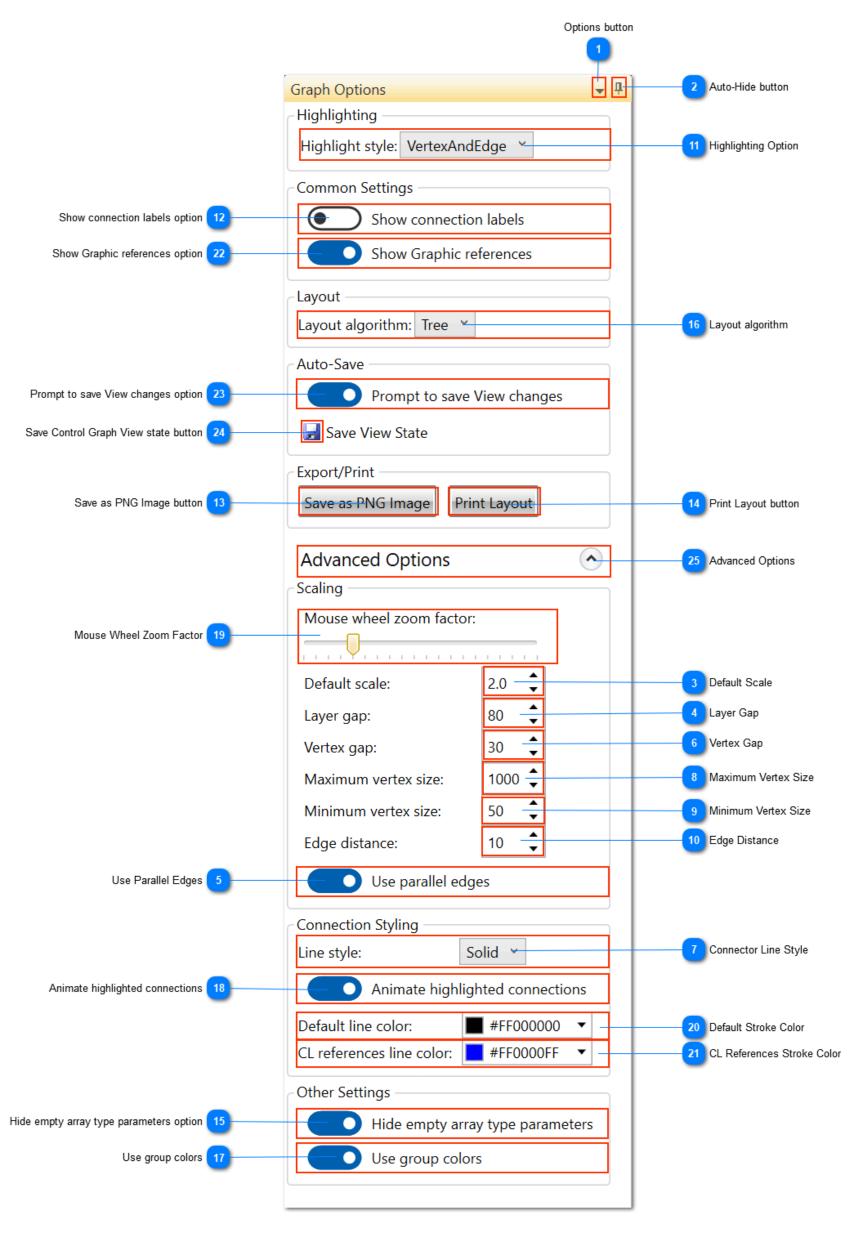


Here is an example of using the Group options when expanding an Entity to delve deeper into its connection relationships. The expanded group has its borders visualized with a unique group color to identify the new expansion group.



# **Control Graph Options**

The following options are configurable that define how visual blocks are laid out on the graph surface and their behaviors when interacting with the mouse.



Options button

Provides docking options for this tool window.

Auto-Hide button



Toggle this button to dock or auto-hide this tool window.

Default Scale

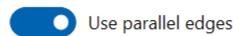


This is the default zoom scale to use when the view is initially displayed. The number of visual blocks that the canvas surface will contain will ultimately determine the initial zoom scale.

Layer Gap
80

This value is used to determine the distance between layers (top-down tree) when laying out the visual blocks.

Use Parallel Edges



Choose this option to show an edge connector for each parameter connection reference. If this option is not checked, then all parameter connection references appear as a single edge connector between to related blocks.

6 Vertex Gap



This value is used to determine the distance between adjacent visual vertices (blocks).

Connector Line Style

Line style: Solid V

Choose the graphical connection style such as solid line, dashed line, etc.

Maximum Vertex Size



This value specifies the maximum block size in the graph layout in pixels. Blocks that reference a large amount of connections such as SCM's will have their size limited to this value.

Minimum Vertex Size

50 🗘

Specify the minimum Vertex size for Vertex blocks that are being sized smaller because of a change in the Edge distance value being changed to a smaller value. This minimum size value only applies to Vertex blocks that have been previously sized to a larger value.

Edge Distance

10 🗘

Specify the distance between edges when the *Use Parallel Edges* option is chosen. Choosing a smaller edge distance value will reduce the size of the Vertex block required to draw the connections.

Highlighting Option

Highlight style: VertexAndEdge Y

You can choose to highlight either the vertex (block), the edge (connector) or both.

Show connection labels option



Choose this option to display the connection label rather than using a mouse hover ToolTip.

Save as PNG Image button

Save as PNG Image

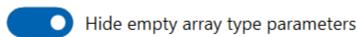
This target button allows you to save the current Control Graph view as a PNG type image file.

Print Layout button

Print Layout

This target button allows you to print the current Control Graph view to a printer device.

Hide empty array type parameters option



Choose this option to hide parameters of type array that have an empty value in the Entity Parameter list.

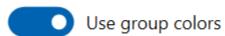
Layout algorithm

Layout algorithm: Tree 🗡

Select the algorithm type that provides the best visual layout of your complex graph.

The EfficientSugiyama and Tree layout algorithms provide the hierarchical type organization. BoundedFR & FR are known as Force Directed algorithms that exhibit symmetries, and tend to produce crossing-free layouts. The LinLog algorithm provides the best visualization for grouped (Entity expansion) graphs.

Use group colors



Choose this option to draw an additional colored border around each Vertex when expanded to visually identify the vertices or entities that were part of the expansion event. Each expansion within the same graph visual will apply a unique color to visually identify multiple expansion groups.

Animate highlighted connections



Choose this option to animate graph edges (connections) when the owning Vertex is hovered over with the mouse. An animated ellipses shape will traverse the path of the graph connection in the direction of the data flow.

Mouse Wheel Zoom Factor

Mouse wheel zoom factor:



Choose the factor to apply in the range of 5-100 to use when zooming the graph contents using the Mouse Wheel. A default value of 25.0 provides a good factor for most graphs. If the zoom is too sensitive, then reduce this factor.

Default Stroke Color

Default line color:



Select the Color to apply to connections in the graph.

🥋 CL References Stroke Color

CL references line color: #FF0000FF ▼

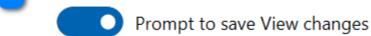
Select the Color to apply to connection types that originate from LCN CL EXTERNAL type references found in AM/MC/PM CL files.

Show Graphic references option



Choose this option to show Graphic connection references on the Control Graph blocks.

Prompt to save View changes option



Choose this option to be prompted to save manual changes to a generated Control Graph view. This prompt will allow you to save the changes prior to closing or navigating to another entity.

The Control Graph view has a Save button in its top Toolbar.



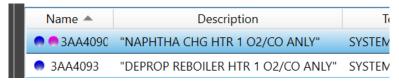
## Save Control Graph View state button



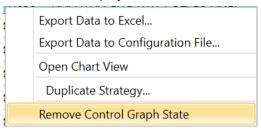
Click this button to save the current Control Graph View state.

The Control Graph View state is saved to the configuration project Database:Collection. If an Entity in the Data View list items has an associated saved Control Graph View state, it will be automatically used when a Control Graph view is requested. You can choose to refresh the restored view to change the Control Strategy exposure.

If an Entity item in the Data View list has a saved Control Graph state, it will be identified using a magenta LED shape in the Name column as shown below.



Control Graph View states can be removed from the configuration project by right-mouse clicking on one or more Entity items in the Data View to display a context menu and select the menu item *Remove Control Graph State*.



Saved Control Graph View states can be preserved in the configuration project when overwriting by choosing the option to "Preserve existing Ink Annotation Files" when importing.



## **Advanced Options**

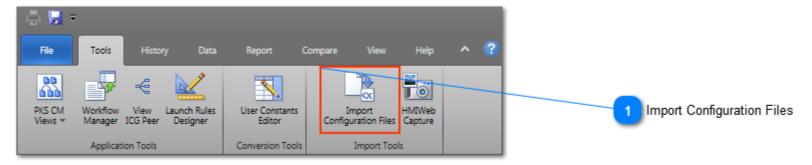
# **Advanced Options**



Click on this expander control to toggle visibility of advanced options.

## Importing Honeywell EB, FSC, SCADA, and Triconex Point Data Sources

## **Overview**



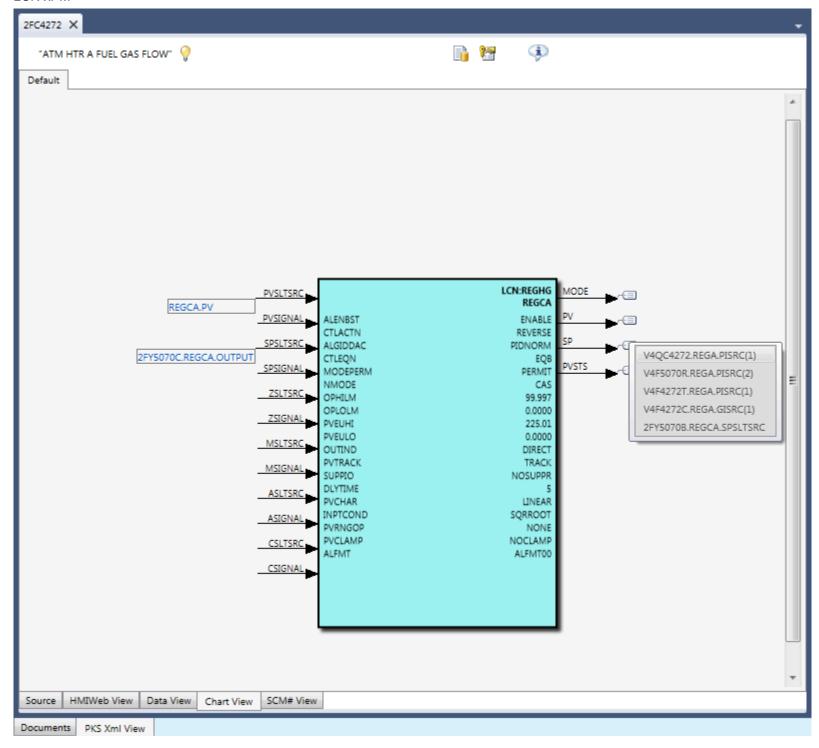
Control Architect supports the conversion of other non-PKS CEE type configuration such as Honeywell Exception Build (EB), FSC (SM), TriStation (TRICONEX), and SCADA Quick Builder to a compliant PKS Configuration standard. Control Architect uses the PKS CEE Configuration schema as the standard for all configuration views and tools.

Entities parsed from EB or SCADA text export files are generated as a simple Control Module with a single embedded Function Block where the embedded block represents the original parsed Entity as in the example below. Block styles and attributes are configurable on a Template or Point type basis. Exported LCN Schematic DX debug text files can be imported to provide graphic Entity reference data in the HMIWeb Data view. Image files can also be imported together with the DX text files to associate a capture graphic image for display within the HMIWeb Data view.

FSC Excel exports are generated as a FLD Control Module with COM and FSC blocks shown as embedded Input/Output blocks.

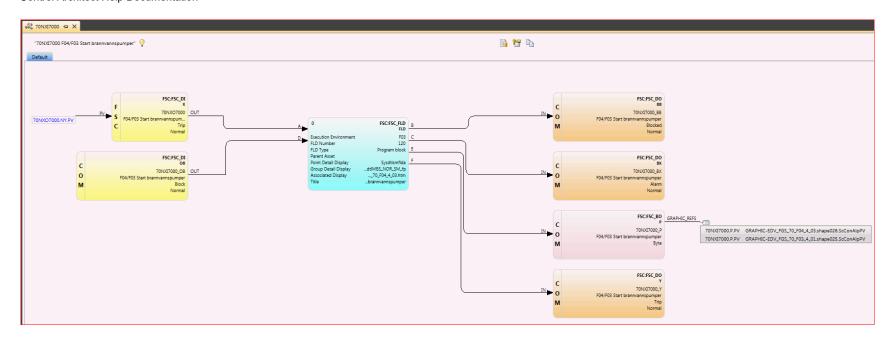
FSC importing supports CDA connection references to existing C300 CEE Control Modules and also FSC to FSC Controller peer connections. HMIWeb graphic references resolve to the FSC FLD block parameters.

#### LCN xPM

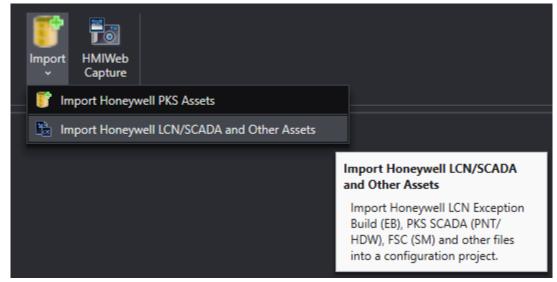


FSC (SM) CDA Integration

Copyright © 2024 by Automation Software Technology, Inc. All Rights Reserved.



# Import Configuration Files



Click this Toolbar drop-down button to launch the tool for importing LCN, PKS SCADA, Triconex, and FSC configuration file based data.

## Working with LCN & SCADA Views

## **Exporting SCADA Configuration from Quick Builder**

Control Architect requires both Point (pnt) and Hardware (hdw) data to be exported from Quick Builder for conversion to PKS standard XML Configuration files. When exporting items from the Quick Builder application select the menu item "File-Export..." and choose the *Scope* option "All items relevant to file type". Expand the "Show Options >>" section and choose the option "Export all Fields" to export all Parameter items regardless of their default value. This will export all configured Channels, Controllers, and Points. A single Point configuration file named [filename].hdw will also be created for all the existing point configuration. Another single Hardware configuration files are then referenced in the conversion tool.

### **Data View Representation of Nodes**

When LCN & SCADA Point types are converted to XML Configuration files their assigned to Node, Box, or Unit is used to form an equivalent PKS Control Execution Environment (CEE) assignment. The example below shows a typical LCN that contains both Hiway, UCN and AM data types that were converted. The data column *AssignedTo* shows how the LCN node types are used to form an assigned to CEE. These nodes and their *AssignedTo* name are used to identify Peer nodes when viewing Peer relationships.

AM type entities are assigned to a node that is formatted using its assigned to LCN Unit id. The example below shows an AM entity 2AC4086X that is assigned to the AM Unit '05' where its *AssignedTo* node is formed using its configured LCN UNIT id. The format used is "AMUNIT\_uu" where uu is the configured LCN Unit id for the entity.

Name 🔺	Description	TemplateName	VersionDate	FunctionBlocks	AssignedTo	ParentAsset
2A5159IB	"VAC HTR	SYSTEM:CONTROLMODULE	04/12/2015 15:13:10	Embedded Function Blocks	UCN_02:NODE_1	"UNIT_06"
2A5161IA	"VAC HTR	SYSTEM:CONTROLMODULE	04/12/2015 15:13:10	Embedded Function Blocks	UCN_02:NODE_1	"UNIT_06"
2A5161IB	"VAC HTR	SYSTEM:CONTROLMODULE	04/12/2015 15:13:10	Embedded Function Blocks	UCN_02:NODE_1	"UNIT_06"
2AA4873	"HC GAS	SYSTEM:CONTROLMODULE	04/12/2015 15:13:11	Embedded Function Blocks	HWY_01:BOX_5	"UNIT_14"
2AA5144B	"VAC HTR	SYSTEM:CONTROLMODULE	04/12/2015 15:13:11	Embedded Function Blocks	HWY_01:BOX_6	"UNIT_06"
2AA5154B	"VAC HTR	SYSTEM:CONTROLMODULE	04/12/2015 15:13:11	Embedded Function Blocks	HWY_01:BOX_6	"UNIT_06"
2AA5220	"DEPROP	SYSTEM:CONTROLMODULE	04/12/2015 15:13:01	Embedded Function Blocks	UCN_02:NODE_9	"UNIT_11"
© 2AA5221	*DEPROP	SYSTEM:CONTROLMODULE	04/12/2015 15:13:01	Embedded Function Blocks	UCN_02:NODE_9	"UNIT_11"
2AA5222	"DEPROP	SYSTEM:CONTROLMODULE	04/12/2015 15:13:08	Embedded Function Blocks	UCN_02:NODE_9	"UNIT_11"
2AA5435	"VENT GA	SYSTEM:CONTROLMODULE	04/12/2015 15:13:11	Embedded Function Blocks	HWY_01:BOX_5	"UNIT_08"
2AC4086X	"Avjet Fre	SYSTEM:CONTROLMODULE	04/12/2015 15:12:29	Embedded Function Blocks	AMUNIT_05	"UNIT_05"
● 2AC4123	"ATM HTF	SYSTEM:CONTROLMODULE	04/12/2015 15:13:11	Embedded Function Blocks	HWY_01:BOX_15	"UNIT_03"
2AC5085A	"ATM HTF	SYSTEM:CONTROLMODULE	04/12/2015 15:13:09	Embedded Function Blocks	UCN_02:NODE_1	"UNIT_03"

#### **Data View Representation of Asset**

LCN Point types use their configured *UNIT* id to form a *ParentAsset*. The ParentAsset is formatted as "UNIT\_uu" where uu is the configured LCN Unit id for the entity.

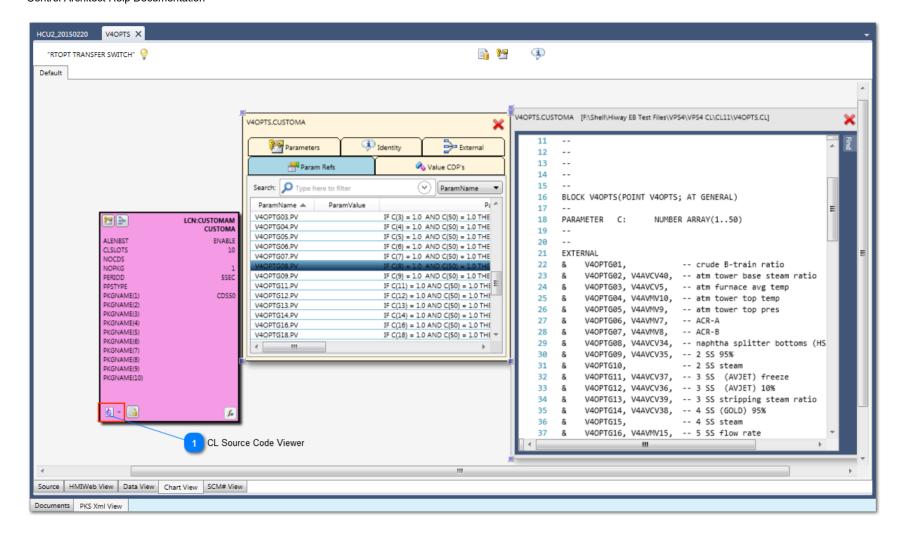
#### **Best Practices for Building LCN, FSC & SCADA Projects**

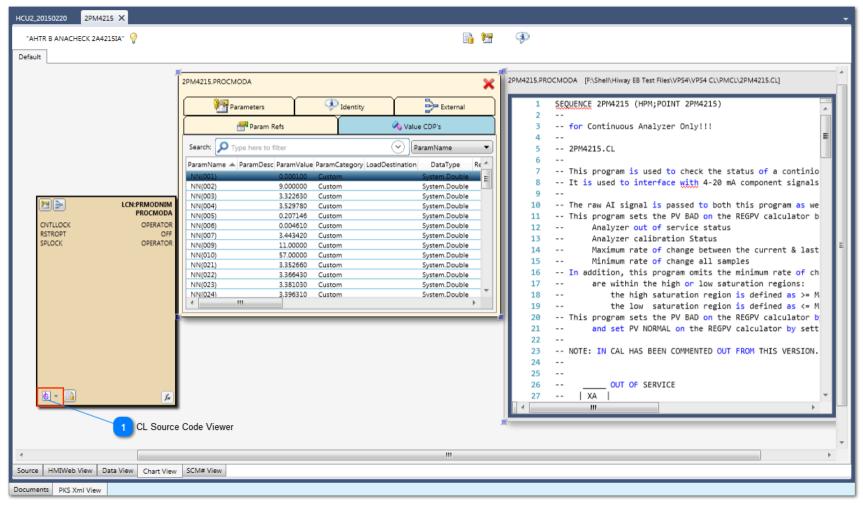
It is recommended that all control related nodes assigned to an LCN are converted to form a single configuration project. Working with a full LCN assures that all connection relationships between AM and other node types are captured along with any UCN node-to-node peer common between xPM and SM node types. Export your LCN Schematic files as debug DX text files for importing into your LCN based configuration project to provide graphic references. If LCN Schematic screen captures are available as Bitmap or PNG file types, then these can also be imported to provide a visual view of the LCN Schematic.

PKS SCADA and FSC type projects should be imported into the configuration project that contains the C300 CEE entities. This assures that any ACE-to-SCADA type OPCServer references are fully identified and navigable through ChartView hyperlink connection references. FSC connection references to C300 CEE functions, FSC-to-FSC Controller peer, and HMIWeb references are supported when FSC data is imported into the same configuration project that contains the C300 CEE entities and HMIWeb graphic files.

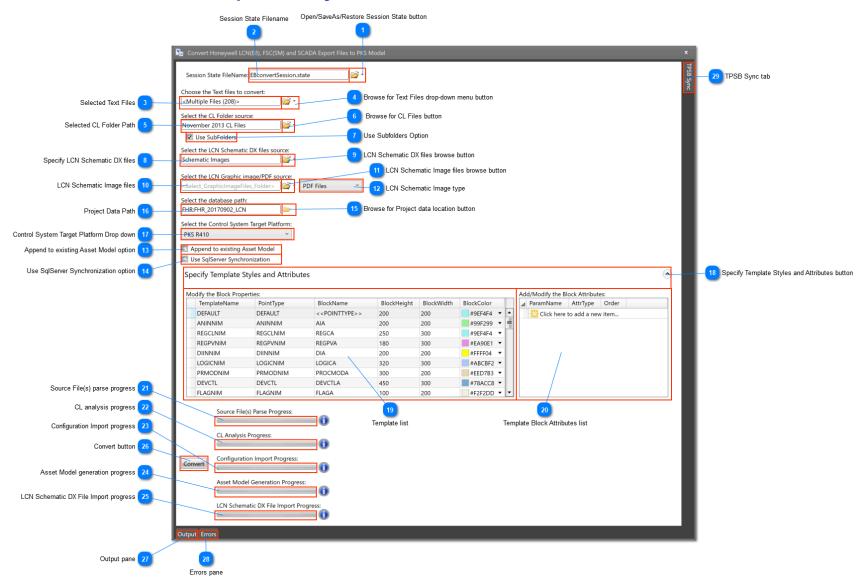
### **How LCN CL Application types are Converted**

LCN point types that support Control Language such as MC/PM/APM/HPM Process Module and AM Bound or Package CL types are converted as a PKS CAB equivalent. The examples below show a CustomAM point type that is configured with a CL Package and an XPM Process Module point type that is bound by a CL Sequence application. The Parameter references found during the CL Analysis phase are created as part of the application definition. Process Module types have their internal parameter types such as NN, FL, TM, STR converted as "Value CDP's".





### Import Honeywell LCN EB and SCADA Point Files



## Open/SaveAs/Restore Session State button



Click this drop-down Menu button to either Open, SaveAs or Restore a Session state. The Restore function allows you to quickly restore the Session state for the last used Session State file displayed in the Session State Filename box.

### Session State Filename

EBconvertSession.state

This is a read-only Edit box that displays the current or last used Session State filename. Specify a filename to save the Session State information for later recall. Typically you will want to maintain a Session State file for your conversion tasks that will allow you to quickly rebuild a conversion project when a new version of configuration is to be updated to an existing XML configuration project.

### Selected Text Files

<Multiple Files (208)>

This is a read-only Edit Box that displays the current set of selected Text files.

## Browse for Text Files drop-down menu button



Click this button to browse for LCN \*.eb or SCADA \*.pnt\\*.hdw or FSC \*.xls Excel files to convert. You cannot convert multiple types in the same import task.

#### Selected CL Folder Path

November 2013 CL Files

This is a read-only Edit Box that displays the selected CL Folder path.

#### Browse for CL Files button



Click this button to browse for the location of your Control Language (CL) files that require analysis for connection relationships to configuration entities.

7

### **Use Subfolders Option**



Choose this option to analyze CL files contained in all subfolders to the selected CL Folder root path.

8

## **Specify LCN Schematic DX files**



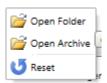
Choose the folder location or an archive file that contains LCN exported Schematic DX files to import. Importing these files will establish graphic references in the HMIWeb Data view.

9

### **LCN Schematic DX files browse button**



Click the drop-down arrow to select either a folder location or an archive file.



Click here to view instructions as to how to create Schematic DX files on the LCN.

10

### **\_CN Schematic Image files**



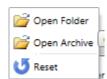
Choose the folder path or archive file that contains image snapshots of LCN Schematics. These imported image files can be shown from within the HMIWeb Data view.

11

### LCN Schematic Image files browse button



Click the drop-down arrow to select either a folder location or an archive file.



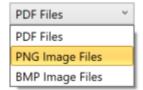
12

### LCN Schematic Image type



Choose the image type to import for referenced LCN Schematic image snapshots.

The supported file types are:



13

### Append to existing Asset Model option

Append to existing Asset Model

Choose this option to preserve existing configuration in the selected Database:Collection that were generated during a previous Import

When a Conversion task is executed a Manifest file is updated that identifies the configuration sources that were used to generate the project configuration.

User will typically first <u>import</u> their PKS Control Builder exported XML configuration files into a specific Database:Collection database. This Import tool would then be used to append to the existing Database:Collection path any other configuration sources such as LCN, SCADA, FSC.



## **Use SqlServer Synchronization option**

Use SqlServer Synchronization

Choose this option to synchronize generated PKS XML configuration from source Honeywell Exception Build (EB) files to a TPS Builder SqlServer Database. This synchronization will generate additional folder (container) Control Modules to add containment relationships.

Copyright © 2024 by Automation Software Technology, Inc. All Rights Reserved.

Browse for Project data location button

FHR:FHR\_20170902\_LCN

Click this button to select or create the Database:Collection where the imported configuration sources will reside.

Project Data Path

This is a read-only Edit box that displays the selected Project Database:Collection path where your configuration sources will be imported to.

Select the Configuration XML Schema to use when importing Configuration source files. During import all configuration sources supported by this Import Tool are converted to a PKS CDA compliant Control Module that will be generated based on the Target schema chosen here.

Specify Template Styles and Attributes button

Specify Template Styles and Attributes

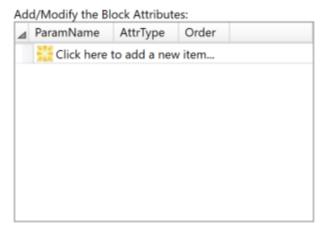
Click this Expander to view the current set of Templates and their Block styles and attributes.

Template list

TemplateName	PointType	BlockName	BlockHeight	BlockWidth	BlockColor
DEFAULT	DEFAULT	< <pointtype>&gt;</pointtype>	200	200	#9EF4F4 ▼ ▲
ANINNIM	ANINNIM	AIA	200	200	#99F299 ▼ =
REGCLNIM	REGCLNIM	REGCA	250	300	#9EF4F4 ▼
REGPVNIM	REGPVNIM	REGPVA	180	300	#EA90E1 ▼
DIINNIM	DIINNIM	DIA	200	200	#FFFF04 ▼
LOGICNIM	LOGICNIM	LOGICA	320	300	#ABCBF2 ▼
PRMODNIM	PRMODNIM	PROCMODA	300	200	#EED7B3 ▼
DEVCTL	DEVCTL	DEVCTLA	450	300	#78ACC8 ▼
FLAGNIM	FLAGNIM	FLAGA	100	200	#F2F2DD ▼ ▼

Displays a list of Template or Point types configured for recognition by this Conversion tool. During the XML file generation operation, the Point type represented by the parsed Entity is matched against a configured Template in this list to determine the styles and attributes to apply to the generated embedded Function Block within the Control Module.

Template Block Attributes list



Displays a list of Block attributes for the selected Block or Template type.

Source File(s) parse progress

Displays the current progress for the parsing operation of the selected Text files.

CL analysis progress

Displays the current progress for the CL file analysis operation.

**Configuration Import progress** 

Displays the current progress for the XML file generation operation.

24 Asset Model generation progress

Displays the current progress for the Asset Model generation operation.

LCN Schematic DX File Import progress
Displays the current progress for the LCN Schematic DX File import operation.

Convert button

Click this button to start or cancel a Conversion task.

Output pane
Output

Click this Tab button to switch to the Output Pane View.

Errors pane

Errors

Click this Tab button to switch to the Errors Pane View.

TPSB Sync tab



Click or hover over this tab to show the <u>options</u> for synchronizing a TPS Builder database to the imported PKS XML configuration data that was converted in this session from Honeywell Exception Build (EB) files. This synchronization generates additional Control Modules to show the same containment hierarchy as found in TPS Builder projects when using "Folder" modules to organize LCN entities.

## **Using Picture Editor MPRINT to create DX Files**

Honeywell LCN document reference "Picture Editor Reference Manual" details use of the MPRINT command to create the Schematic DX debug files that can be imported into Control Architect for providing entity references and mapping between LCN schematics and configuration entities.

### Picture Editor Reference Menu excerpt

3.4.3 Multiple Print Command Command: MPRINT <pathname>

Abbreviated forms: MPRN <pathname>

MPR <pathname>
MP <pathname>

where <pathname> is a pathname to a Schematic List file.

Use—The Multiple Print command accesses a file that contains a list of schematic source files. It then attempts to generate a schematic print file for each source file named on the list. There is no output to the printer. In addition, it generates a file that contains the result of each print request. The results file is created in the same directory as the schematic list file. The print file contains a list of objects and text in the picture, including their coordinates, colors, priorities, behaviors, and target actions.

Schematic List File—Before executing the MPRINT command you must build a Schematic List file with the Text Editor. The file name can contain up to 8 characters and must end with a .EL suffix. EL stands for edited list. While using the Picture Editor, you can press the ESC key to temporarily use the Text Editor. Hold down CTL and press HELP when you are ready to return to the Picture Editor.

REFERENCE: For details on the Text Editor, refer to the Text Editor Operation manual. Note that when the MPRINT command is executed, the Default Pathname is initialized to that path specified for the Schematic List file. As explained below, this may be important if files specified on the list use a partial pathname.

Within the Schematic List file:

Any line beginning with { (a left curly brace) is treated as a comment line and is not executed. If a comment requires more than one line, each line must begin with {to be treated as a comment. A right curly brace (}) is optional at the end of a comment line. Schematic source file names can be specified with a full or partial pathname. The .DS extension is optional. If a partial pathname is specified, it is used in conjunction with the Default Pathname. Specification of a full pathname reinitializes the Default Pathname.

An example of a Schematic List file follows:

\$F2>WORK>SCHEM1 NET>HMV1>SCHEM2.DS

{The following file should execute as NET>HMV1>SCHEM3.DS} SCHEM3.DS
{The next file should execute as NET>HMV1>SCHEM4.DS} SCHEM4

Picture Editor Reference Manual 39 08/2016 {Use of the right curly brace on comments and addition {of the .DS extension on source files is optional.

**Procedure**—On the command line, type MPRINT followed by a full or partial pathname for the Schematic List file, then press the ENTER key.

For example: MPRINT NET>HMV1>SLIST

The Default Pathname is modified to use the supplied name. Refer to the Set Pathname command for acceptable forms of the pathname. Press ENTER to continue or press CANCEL to cancel the command. After pressing ENTER, execution begins and continues until each file on the list is processed.

If you wish to halt before the operation is complete, hold down CONTROL and press BREAK.

Copyright © 2024 by Automation Software Technology, Inc. All Rights Reserved.

After the current file is processed, the operation is aborted.

**Options**—within the Schematic List file, several options can be added following the Schematic Source file names:

- APPEND will append the printed information to the existing print file for that source file.
- COMMENT will print only the comment sentences for that source file.
- DEBUG will include the debug sentences for that source file.
- SYMBOLS will print only the symbols sentences for that source file.

For example: NET>HMV1>SCHEM1 DEBUG

**Errors**—If an error occurs during execution, a print error is placed in the Results file and processing of the list file continues.

If the storage media becomes full during execution of the Multiple Print command, a print error is placed in the Results file for that schematic name and all remaining schematic names in the list file. If the Results file cannot be written to the storage media, then an error message appears on the screen.

Each Schematic Print file is generated in the same directory as its respective schematic source file. The Print file has the same name as the Schematic Source file, but has a .DX suffix. Each Schematic Print file is ASCII formatted. Any Schematic Print file with the same name that existed before executing the Multiple Print command is overwritten (except see the Append option).

**Results File**—A Results file is produced by the Multiple Print command. It is generated in the same directory and with the same name as the Schematic List file except it has a suffix of .ER. Press the ESC key to temporarily use the Command Processor. Use the Print command to view the Results file. For example, PR Net>HMV1>SLIST.ER.

The Results file contains one line for each executable file name specified in the Schematic List file. Each line shows the full or partial file pathname and a print status. List file comments are Picture Editor Reference Manual 40 08/2016

not included. The .DS suffix does not appear in the Results file even if it was used in the Schematic List file.

If a previous Results file with the same file name exists, it is overwritten.

Only the first 30 characters of a defined pathname for each line in the list file appear in the Results file.

If the Multiple Print operation is halted by pressing the CONTROL-BREAK keys, the Results file contains the message: Aborted by User Request.

An example of a Results file is:

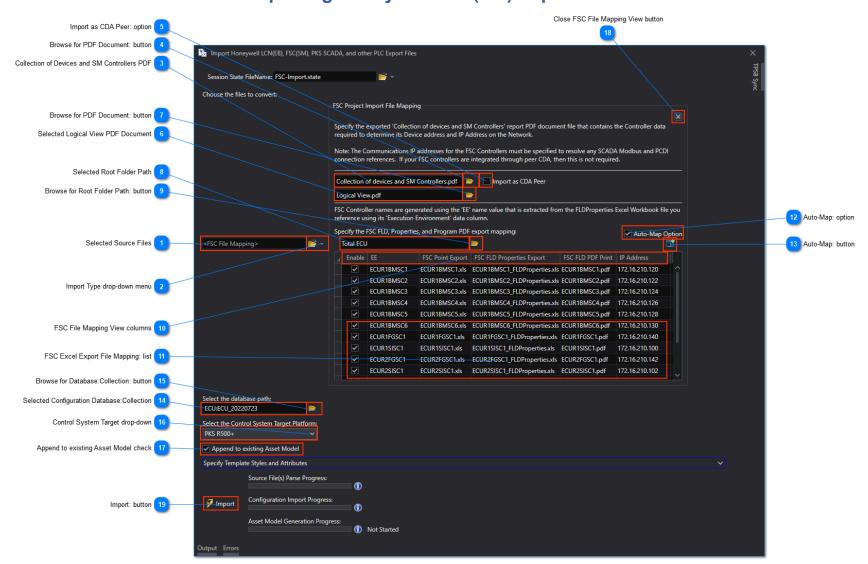
```
$F2>WORK>SCHEM1 Success!
NET>HMV1>SCHEM2 Print Operation Failed
SCHEM3 Read Operation Failed
SCHEM4 Success!
```

To view the individual schematic print files, escape from the Picture Editor, then use the Command Processor functions to view each file. For example:

```
PR NET>HMV1>SCHEM1.DX.
```

If you precede the print command with a Data Out command to the printer (DO \$Pn), you can obtain a paper copy.

## Importing Honeywell FSC(SM) Export Files



## Selected Source Files

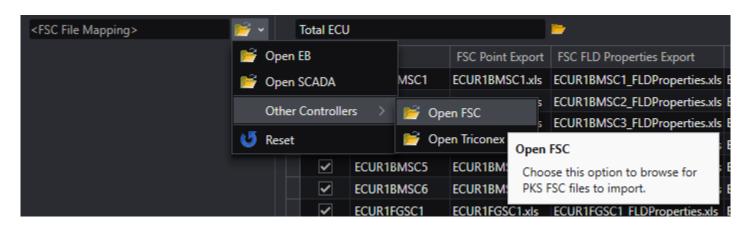
<FSC File Mapping>

This edit box will indicate <FSC File Mapping> when FSC is chosen as the Import type.

# Import Type drop-down menu



Choose this menu item to import FSC (SM) type export files.

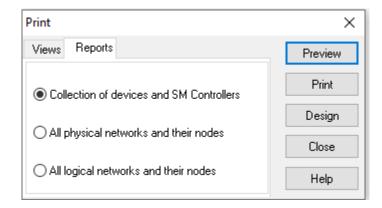


### **Collection of Devices and SM Controllers PDF**

Collection of devices and SM Controllers.pdf

Click the Browse button to select the SM exported PDF Document that represents the "Collection of devices and SM Controllers".

From the Safety Builder "Network Configuration" view click the Print function and select the report to print:



## Browse for PDF Document: button



Click this Browse button to select the SM exported PDF Document that represents the "Collection of devices and SM Controllers".

## Import as CDA Peer: option



Choose this option to import configuration data that represents FSC Controllers that are configured for CDA integration into a C300 Process controller.

If your FSC Controllers are communicating with your C300 through Modbus configuration such as Scada and PCDI, then do not choose this option. Additional data is required to resolve Modbus connected FSC controllers such as IP addresses and the Logical View document that defines the PCDI Device address.

### Selected Logical View PDF Document



Click the Browse button to specify the Logical View PDF Document.

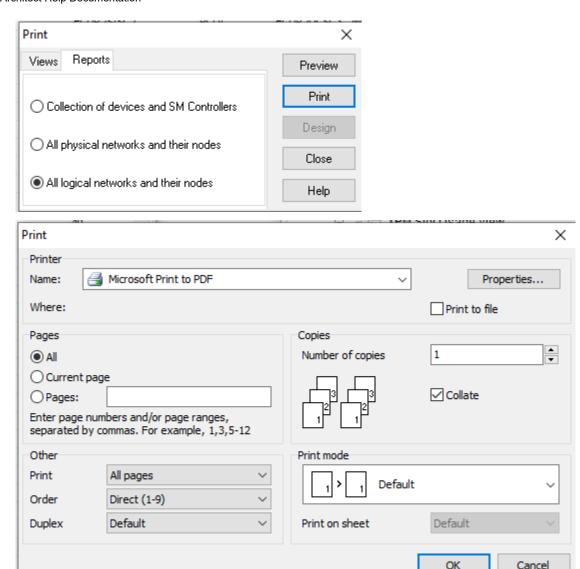
This document provides the necessary PCDI communications binding between the FSC Controller and the DCS C300 Controller along with its configured PCDI Device ID. This data is used to resolve the PCDI Modbus addresses used to communicate with the FSC Controller Application FLD logic.

## Browse for PDF Document: button



Click this Browse button to specify the Logical View PDF Document.

From the Safety Builder "Network Configuration" view click the Print function and select the report to print to a PDF document as shown below:

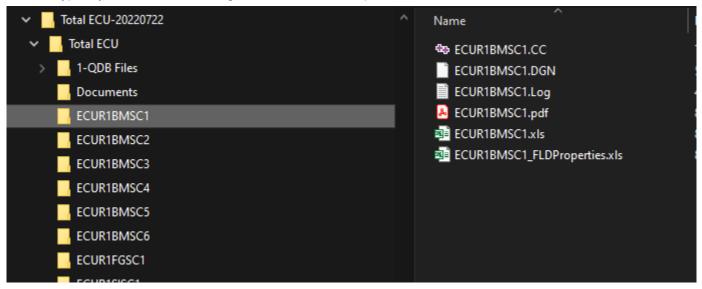


# Selected Root Folder Path

Total ECU

Click the Browse button to select the root folder where your FSC exported configuration documents reside.

This will typically look like the following where the root folder path would be "Total ECU":



### Browse for Root Folder Path: button



Click this Browse button to select the root folder where your FSC exported configuration documents reside.

# FSC File Mapping View columns



- Enable Check this column for each item that is to be imported and resolved. If the IP Address used for the Modbus communications is not provided, then you cannot enable the item.
- EE the execution environment value read from the FSC FLD Properties Excel export file
- FSC Point Export the excel export file that contains the FSC FLD point data { FLD blocks }
- FSC FLD Properties Export the excel export file that contains the FSC FLD properties data { FLD definition, ParentAsset, etc. }
- FSC FLD PDF Print the printed FSC Controller Application FLD logic drawings

• IP Address - the Modbus IP Address that is used to communicate between the FSC Controller and the PKS Scada Controllers. This address is parsed from the "Collection of Devices & SM Controllers" PDF document. If the IP address fails to be parsed, then the user must manually provide this IP address.

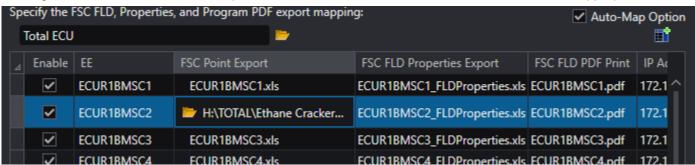
## **FSC Excel Export File Mapping: list**

~	ECUR1BMSC6	ECUR1BMSC6.xls	ECUR1BMSC6_FLDProperties.xls	ECUR1BMSC6.pdf	172.16.210.130
<b>✓</b>	ECUR1FGSC1	ECUR1FGSC1.xls	ECUR1FGSC1_FLDProperties.xls	ECUR1FGSC1.pdf	172.16.210.140
~	ECUR1SISC1	ECUR1SISC1.xls	ECUR1SISC1_FLDProperties.xls	ECUR1SISC1.pdf	172.16.210.100
<b>✓</b>	ECUR2FGSC1	ECUR2FGSC1.xls	ECUR2FGSC1_FLDProperties.xls	ECUR2FGSC1.pdf	172.16.210.142
<b>✓</b>	ECUR2SISC1	ECUR2SISC1.xls	ECUR2SISC1_FLDProperties.xls	ECUR2SISC1.pdf	172.16.210.102

Displays the current FSC Excel export file mapping.

When the user maps the FSC FLD Properties Export file, the CEE assigned value is parsed from the Excel file and displayed as the EE value for the mapping.

Clicking in a row in the DataGrid will expose a File Browse button that can be clicked to select the appropriate Excel/PDF export file.



## Auto-Map: option



Choose this option to allow auto-mapping of your FSC exported configuration documents.

This auto-map function depends on the selected root folder path. All subfolders to this root folder path are auto-browsed to discover any exported Excel property documents and a single printed Application FLD PDF documents. The naming of these 3 exported/printed documents must be similar as shown where for example the FSC Controller ECUR1BMSC2 would contain an Excel point export document named *ECUR1BMSC2.xls*, an Excel FLD Properties export document named *ECUR1BMSC2.pdf*.

When exporting the Excel documents from the Honeywell Safety Builder application, these will become the default names that are chosen. The application FLD program logic print to PDF will require you to provide the exact same name as the FSC Controller with the pdf extension. Failure to follow this naming schema will prevent the auto-map function from properly locating and mapping these documents.

### 😱 Auto-Map: button



Click this button to execute the Auto-Map function.

If existing FSC Controller mappings exist, you will be prompted to clear these existing items. If you choose Yes, then existing items will be cleared and new mappings will be added. If you choose No, then only new mappings that were discovered will be added to this list.

### Selected Configuration Database: Collection



Displays the selected Database: Collection where the imported FSC data will be created.

### Browse for Database:Collection: button



Click this button to select an existing or create a new Database: Collection in your connected MongoDB database.

### Control System Target drop-down



Choose the supported PKS System release that defines the type of XML configuration files that are created and imported.



## **Append to existing Asset Model check**

✓ Append to existing Asset Model

When importing into an existing configuration project that contains C300 CEE Control Modules, choose this option to append the FSC imported data. Failure to do so, will delete existing configuration data.

18

## **Close FSC File Mapping View button**



Click this button to close the FSC File Mapping View. You can reopen this view by again choosing the *Open FSC* from the Import type drop-down menu.

19

### **Import: button**



Click this button to begin importing your FSC (SM) Controller configuration data.

## **Getting Started using FSC(SM) Integration**

Control Architect supports the importing of FSC(SM) exported project configuration data for integration of PLC Modbus data through PKS PCDI Channel blocks and PKS SCADA point blocks. The import process imports exported Safety Builder documents and generates Control Modules that contain a single embedded block to represent a PLC Controller and blocks to represent PLC Logic Programs. Users can now see the Modbus address and PLC Alias name being referenced in their Control Strategies when viewed in a Chart View. A Program Logic viewer allows users to view the Logic as a PDF view. A Variable list allows users to navigate to its usage in the Program Logic.

FSC Controllers that purely integrate into a C300 CDA using gateways are imported such that a FLD Peer Control Module will be created for each FLD that is integrated into the C300.

### **Prepare PKS Configuration Snapshot**

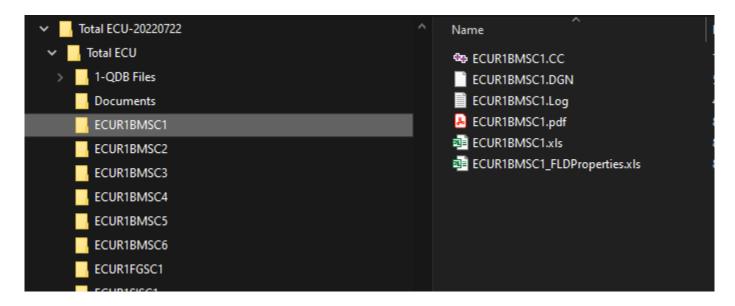
A new PKS configuration Collection snapshot must be created using a minimum Control Architect release of 21.11.3. Beginning with this release during the import of PKS Server configuration special processing is added for PCDI Channel Block connections to support PLC Modbus connectivity. Existing configuration snapshots created with an earlier release are not supported for PLC Modbus connectivity.

If your configuration project consists entirely of SCADA, then having to rebuild existing configuration snapshots is not required as there are no additional connection processing required for SCADA points.

### **Exporting data from Safety Builder application**

#### **Building the Project Export Folder Content**

The data required for importing into a configuration project Collection snapshot must reside in a folder hierarchy similar to below. The root folder path "Total ECU" is then referenced in the Import tool when choosing to import FSC type PLC configuration.



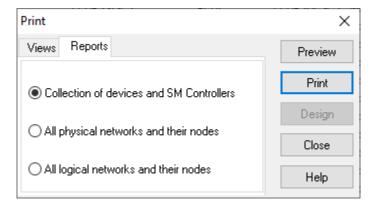
#### **Collection of devices and SM Controllers Document**

To begin importing Safety Builder exported data sources, the "Collection of devices and SM Controllers" report must be printed to a PDF document type. This document will be read to extract its Project Information and Modbus Communications IP Address data. These properties are used to generate a unique PLC Controller Strategy name and these property values are cached in this new PLC Controller Strategy to support future version comparison operations. Modbus Communications IP Address data is also parsed to allow resolving of PKS Scada Controller and its point Modbus addresses.



This referenced document is not required when choosing to import Safety Builder configuration that is purely CDA integrated using gateways.

The "Network Configurator" view in the Safety Builder must be selected to print this document.



### All logical networks and their nodes Document

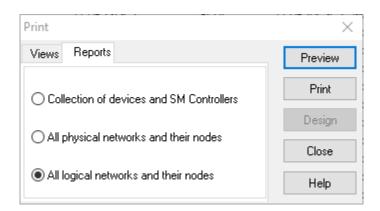
Copyright © 2024 by Automation Software Technology, Inc. All Rights Reserved.

The "All logical networks and their nodes" configuration document needs to be printed as a PDF document type. You can choose to name this anything as you will be required to browse for this particular PDF document file. This document provides the PCDI Communication bindings and the PCDI Device address.



This referenced document is not required when choosing to import Safety Builder configuration that is purely CDA integrated using gateways.

The "Network Configurator" view in the Safety Builder must be selected to print this document.



#### **Application FLD Logic PDF Documents**

All Program FLD Logic documents must be exported from Safety Builder as PDF document types using the File-Print menu item. This will display a "Print" window that allows you to choose the "All" option. While printing to a PDF the user will be prompted to provide a SaveAs PDF filename, which must be the exact name of the FSC Controller that contains this FLD logic. During import these Program FLD Logic documents will be imported to the MongoDB configuration snapshot where these FLD logic drawings can be viewed using the PLC Logic viewer. Program Logic strategies are also created during import that allow the user to launch a Program Logic viewer window for viewing logic and for navigating its used variable list.

The "Application Editor" view in the Safety Builder must be selected to print this document.

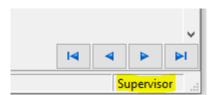
#### Excel (\*.XLS) Data Files

The following Excel documents must be exported from Safety Builder as Excel \*.XLS types.

- Point Properties xls
- FLD Properties xls



The Safety Builder instance must have its privilege level elevated to export the "FLD Properties" document when the "Application Editor" view is selected. This is changeable by left mouse double-clicking in the bottom right pane of the Safety Builder application window.



Export these documents from Safety Builder by:

- 1. Select the "Point Configurator" view button
- 2. Click the "Export Points" button from the main ToolBar or from the File-Export Points... menu item
- 3. Use the default name provided, which will become the name of the FSC Controller and the xls file extension. The save location should default to the subfolder that is named from the active FSC Controller.
- 4. Select the "Application Editor" view button
- 5. Select the menu item File-Import and Export- Export FLD Properties... menu item.



You must have elevated permission to export this document.

6. Use the default name provided, which will become the name of the FSC Controller with "\_FLDProperties" appended to the filename along with the xls file extension. The save location should default to the subfolder that is named from the active FSC Controller.

## Importing Safety Builder exported data into a Control Architect configuration project

## **Specify the Collection of devices and SM Controllers Document**

Copyright © 2024 by Automation Software Technology, Inc. All Rights Reserved.

Control Architect Help Documentation

The "Collection of devices and SM Controllers" PDF document must be available for browsing. You can choose to name this file using any valid filename.

This document is used to provide project information and Modbus IP Address data during import. The project information is used to provide description and last saved property data when generating the PLC Controller Control Module.

#### Specify the All logical networks and their nodes Document

The "All logical networks and their nodes" PDF document must be available for browsing. You can choose to name this file using any valid filename.

This document is used to provide Modbus PCDI bindings and Device Address data during import.

#### **Program FLD Logic PDF Documents**

Application FLD logic PDF documents are imported into your configuration snapshot where they can be viewed using the PLC Viewer.

#### **Modbus Declaration Map**

During import a mapping of all variables defined in the FSC Controller is created that also specifies where the Variables are used along with their Modbus Alias name, Modbus Address, and its PCDI Modbus address if used as a PCDI reference. This mapped document is used to resolve PCDI Channel and SCADA point block Modbus address references during the import operation. This map is also saved to the generated PLC Controller CM Strategy as a JSON type data collection.

#### **Possible Errors when Resolving Modbus Connectivity**

It is expected that no errors will occur during import, but reality is that at times what is configured in the DCS won't necessarily match what the PLC project has defined. For example a SCADA point strategy could be referencing a Modbus Address that is no longer defined in the PLC project where the SCADA point should have been either deleted or updated to a new address. Failure to resolve these Modbus connections will be noted as an error in the Error tab view. Errors listed in this view are often indicative of a defect in the configuration.

Modbus references that could not be resolved will be displayed as an invalid Modbus reference.



### **Import results**

Upon successful completion of an Import operation documents will have been imported into a MongoDB Database Collection that is logically named "PLCAssets". The documents that were imported consist of the Program Logic PDF files, Modbus Alias mapping, and Excel files.

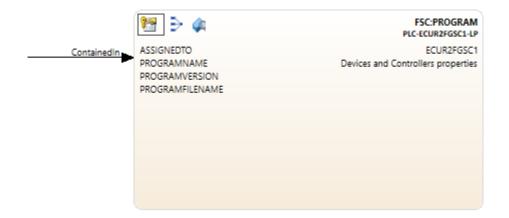
#### **Control Module Strategies that are created during Import**

During import a number of strategies that represent both the PLC Controller node and its contained PLC logic programs are generated as Control Modules to fit within the PKS Control Strategy schema. Program Logic CM's are contained within the PLC Controller CM. Users can view the assigned and containment relationships by choosing to group the configuration Data View.

A single Control Module is generated to represent the PLC Controller node itself.



Control Modules that represent FSC PLC Program Logic are named using the format: *NodeName* where *NodeName* is the name of the FSC Controller node.



### **Overwriting Existing FSC Import**

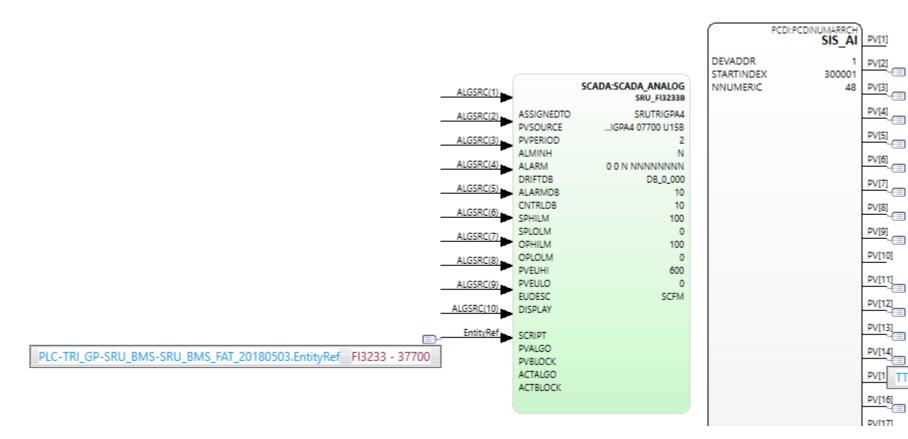
Users can choose to overwrite an existing import. If the imported PLC Controller Strategy name is the same as an existing, then it along with all its documents and generated Control Module strategies will be replaced or overwritten. Connections that were generated to support SCADA and PCDI relationships for the PLC Controller being imported are deleted prior to reimporting.

### **ChartView Visualization**

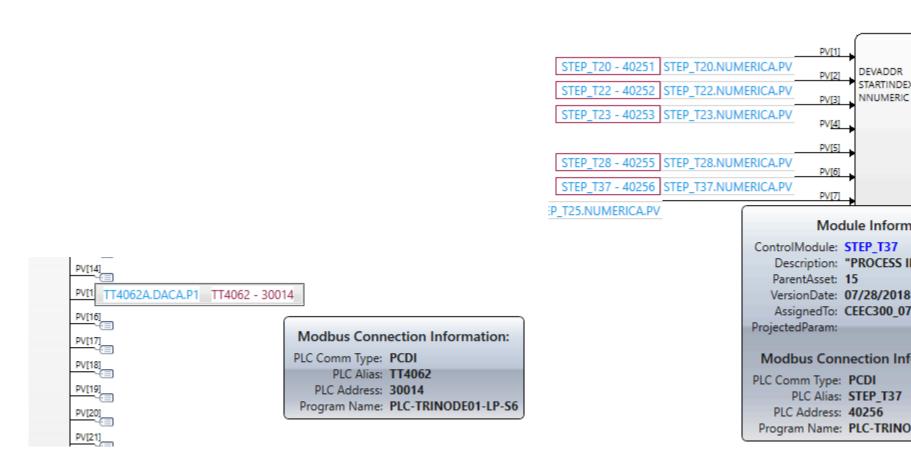
The Chart View has been enhanced to support the visualization of a Modbus connection type. These connection types are expected for PCDI Channel and SCADA point blocks.

### **Visualization of Modbus Connection types**

Here is an example of what to expect when viewing PCDI Channel blocks or SCADA point blocks within a Chart View.



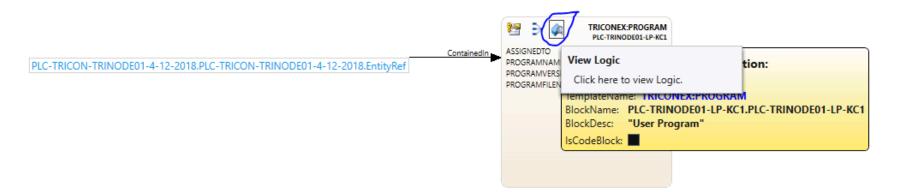
When hovering your mouse over a Modbus connection type, additional information is displayed.



#### **Viewing Program Logic**

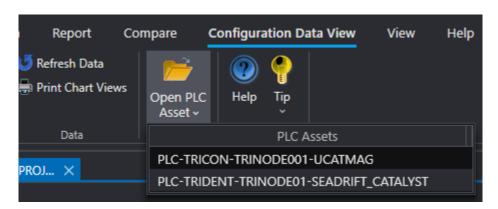
The Program Logic associated with a PLC Program Logic block can be viewed by clicking a button that becomes visible in the adorner layer of a Chart View block when the mouse enters a block visual.

When this View Logic button is clicked a Program Logic View window will be launched.

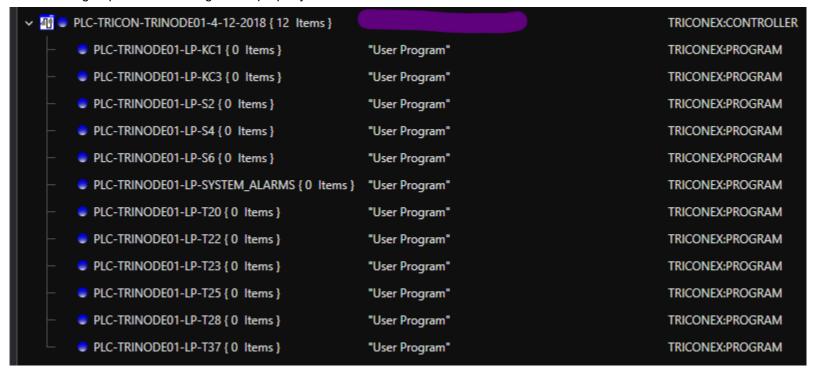


Program Logic can also be viewed from the Data View tab. The Ribbon Toolbar for the Data View tab contains a drop-down button that lists all PLC Assets that are discoverable in the current project configuration snapshot. This viewer allows you to view all logic programs and their variable usage within a single viewer unlike the Program Logic block that is scoped to only that PLC program itself.

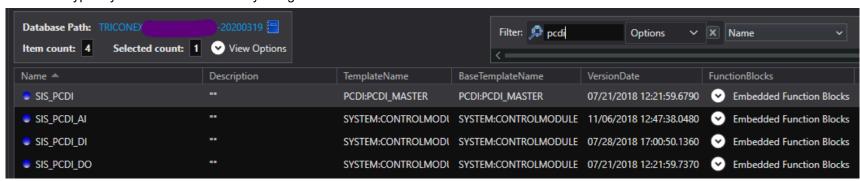
Copyright © 2024 by Automation Software Technology, Inc. All Rights Reserved.



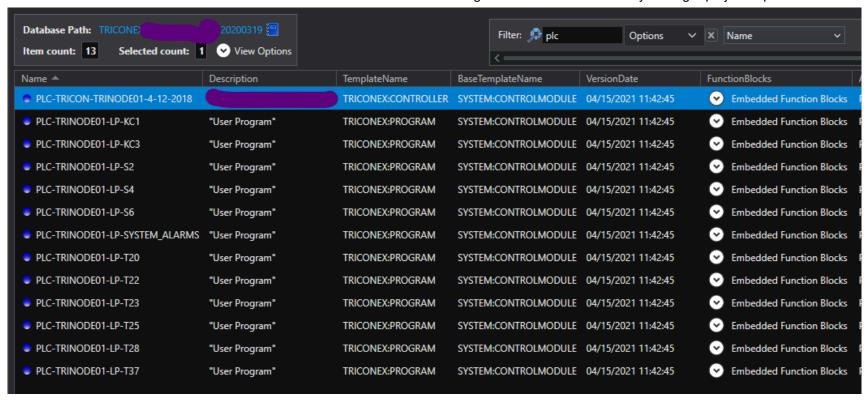
Use the grouping functions to see the containment relationship between a PLC (FSC) Controller block and its Program Logic blocks. This view is grouped on the "AssignedTo" property.



Users can typically view their PCDI data by using the filter in the Data View to filter on Name or FunctionBlocks with the filter text "PCDI".



Users can also filter on the text "PLC" to achieve a similar related view of strategies associated with a Safety Manager project import.



### **Viewing the Manifest Data**

The Manifest Data that is created and maintained for each configuration snapshot provides history and information as to the contents of a configuration snapshot. This view is accessible from the configuration <u>Data View</u>. This Manifest View provides the functions that allow you to attach or create a new binding between an existing PLC project import and the current configuration snapshot.

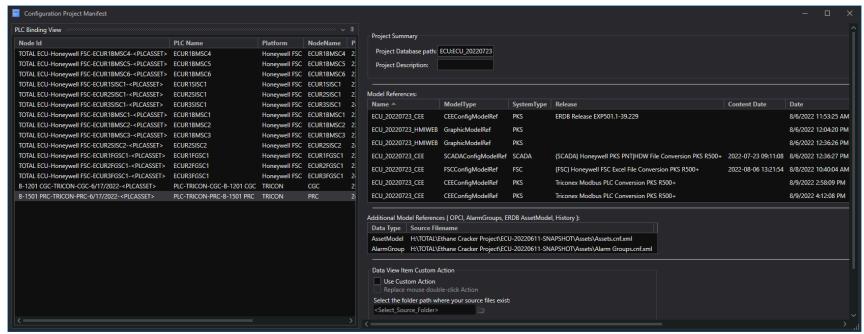
#### PLC Binding to a Configuration Project Database: Collection Snapshot



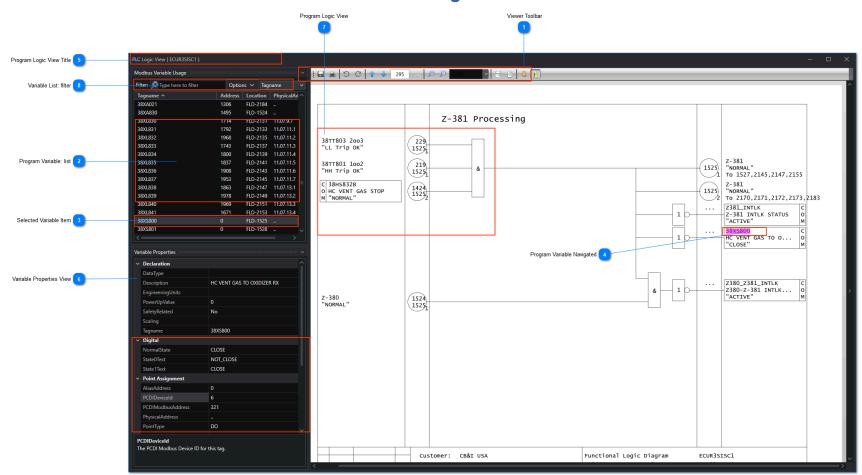
A future release will support the ability to bind an already imported Modbus asset to multiple Database:Collection configuration snapshots.

When a new configuration snapshot is created the user can choose to manually bind an existing PLC project entity to this new configuration snapshot rather than having to re-import the same PLC project information. This assumes that the PLC project configuration itself has not changed.

The existing Manifest View has been enhanced to support a new "PLC Binding View" Tool window. This view provides any future maintenance functions that will become available for working with imported Modbus assets.



### **FSC PLC Logic View**



# Viewer Toolbar



Click the button in this Toolbar to Save, Print, Navigate, Zoom, etc. the view content.

## 🦱 Program Variable: list

38XL830	1714	FLD-2131	11.07.9.7
38XL831	1792	FLD-2133	11.07.11.1
38XL832	1968	FLD-2135	11.07.11.2
38XL833	1743	FLD-2137	11.07.11.3
38XL834	1800	FLD-2139	11.07.11.4
38XL835	1837	FLD-2141	11.07.11.5
38XL836	1908	FLD-2143	11.07.11.6
38XL837	1953	FLD-2145	11.07.11.7
38XL838	1863	FLD-2147	11.07.13.1
38XL839	1978	FLD-2149	11.07.13.2

Displays a list of variables that are used in this PLC Program Logic. Mouse double-clicking a list item will navigate to the variable usage in the program and highlight it using the Chart View NonCEE reference defined color. Variable that have multiple usages as defined by multiple sheet locations can be navigated sequentially by repeated mouse double-click actions on the variable item.

### Selected Variable Item



Program variables used in this program show their Program Name, Variable Declaration Name, and Sheet locations data.

### Program Variable Navigated

#### 38XS800

When a program variable in the Variable list is mouse double-clicked, this first items sheet location is brought into view and the found item is highlighted.

### 🤁 Program Logic View Title

#### PLC Logic View { ECUR3SISC1 }

Logic programs are displayed in their own Program Logic View. Views can be identified by the title that is displayed in the Window Title area at the top left of a window.



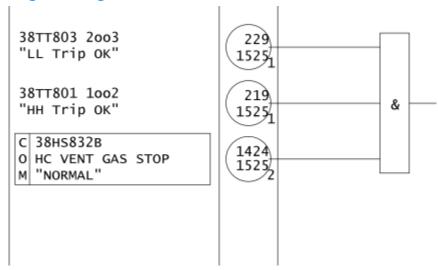
### **Variable Properties View**



Displays the properties for the selected Program Variable.



### **Program Logic View**



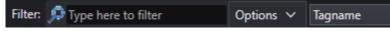
Displays the Program Logic that was exported from the PLC application and imported into your configuration Collection snapshot as a PDF document type.

The Toolbar provides zoom, page navigation, saving, printing, and window fit functions.

Mouse double-clicking on a Program Variable item in the list will select and bring into view where the Variable is being used. The navigated item is highlighted in a magenta color. Variables that are used in multiple locations in the program logic can be navigated to its next location by again executing a mouse double-click action. If you subsequently double-click to navigate again, but there are no more locations, then the current highlighted program variable has its highlight cleared.

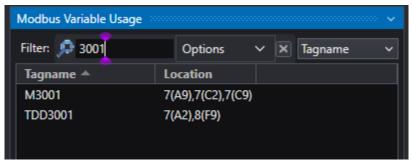


### Variable List: filter

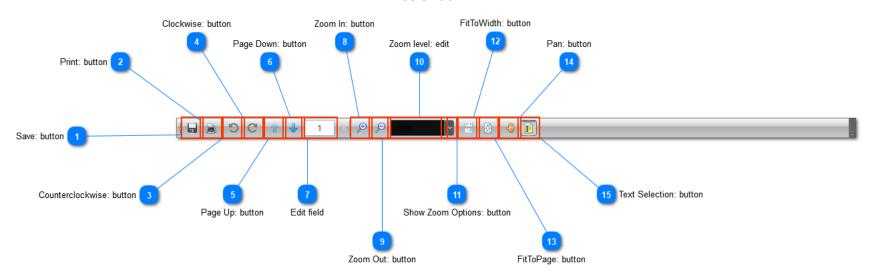


Enter filter text to filter the list of Modbus variables.

The list is now filtered on any Tagname that contains the text phrase "3001". Click the Clear 'x' button to clear the filter text and the filtered list.



### toolbar



Save: button



Click this button to save the Program logic to a PDF document type.

Print: button



Click this button to print the Program logic.

Counterclockwise: button



Click this button to rotate the Program logic view counter clockwise.

Clockwise: button



Click this button to rotate the Program logic view clockwise.



Click this button to navigate to the previous page.





Click this button to navigate to the next page.

Edit field



Enter a page number to navigate to.

Zoom In: button



Click this button to zoom into the Program logic view.

**Zoom Out: button** 



Click this button to zoom out of the Program logic view.

Zoom level: edit



Enter the zoom level.

11

## **Show Zoom Options: button**



Click the drop-down button to select a zoom level.

FitToWidth: button



Click this button to change the view to fit to the current width.

FitToPage: button



Click this button to change the view to fit to the full page size.

Pan: button



Click this button to pan using the mouse cursor.

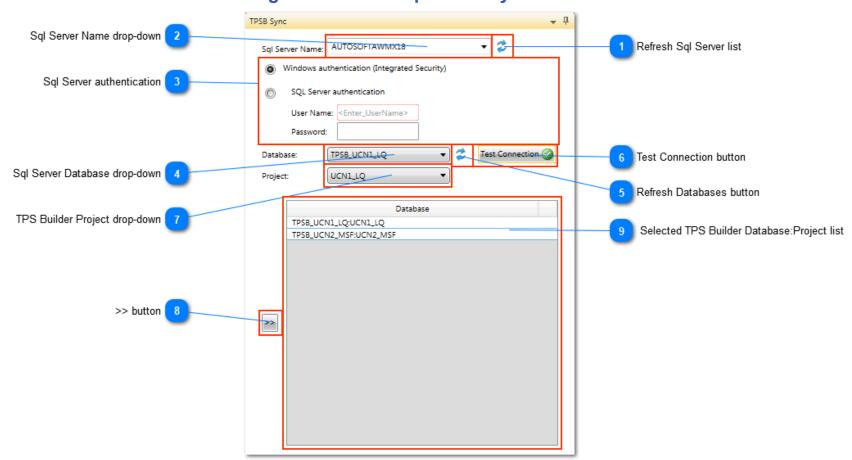
When a program variable item is navigated from the Modbus Variable list, the mode is changed to "Selection".

Text Selection: button



Click this button to select text using the mouse cursor.

## **Using TPS Builder SqlServer Synchronization**

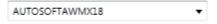


# Refresh Sql Server list



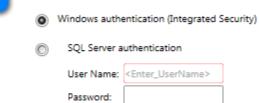
Click this button to refresh the list of available SqlServer instances found.

# Sql Server Name drop-down



Select a SqlServer instance from this drop-down list or enter the SqlServer instance name to connect with.

## Sql Server authentication



Choose Integrated Security or provide logon information to connect with the selected SqlServer instance.

## Sql Server Database drop-down



Select a Database from the SqlServer to connect with.

## Refresh Databases button



Click this button to refresh the list of available databases in the selected SqlServer instance.

#### Test Connection button



Click this button to validate SqlServer connectivity using the selected TPS Builder Database and Project.

### TPS Builder Project drop-down



Select a TPS Builder Project to connect with.



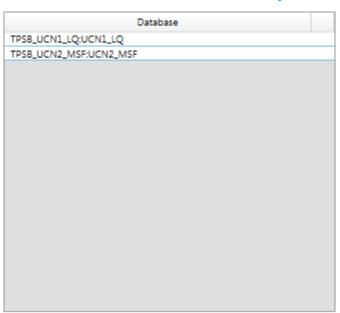
### >> button



Click this button to add the selected SqlServer Database and TPS Builder Project to the list of databases to use for synchronizing a generated PKS XML configuration project.



## Selected TPS Builder Database:Project list

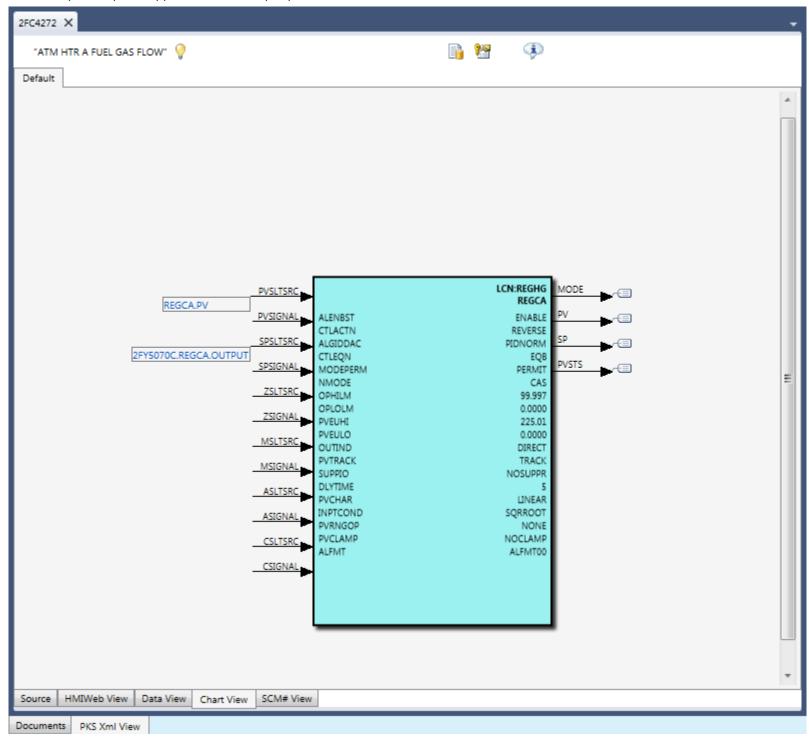


Specify the TPS Builder projects that will be used to synchronize generated PKS XML configuration.

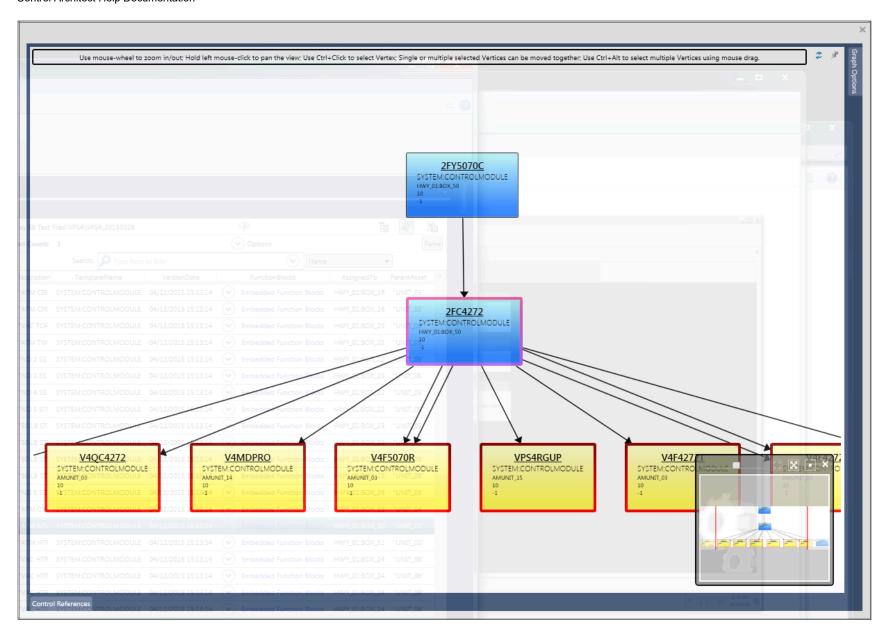
# **Example Configuration Views**

Below are example views for various LCN and SCADA type configuration.

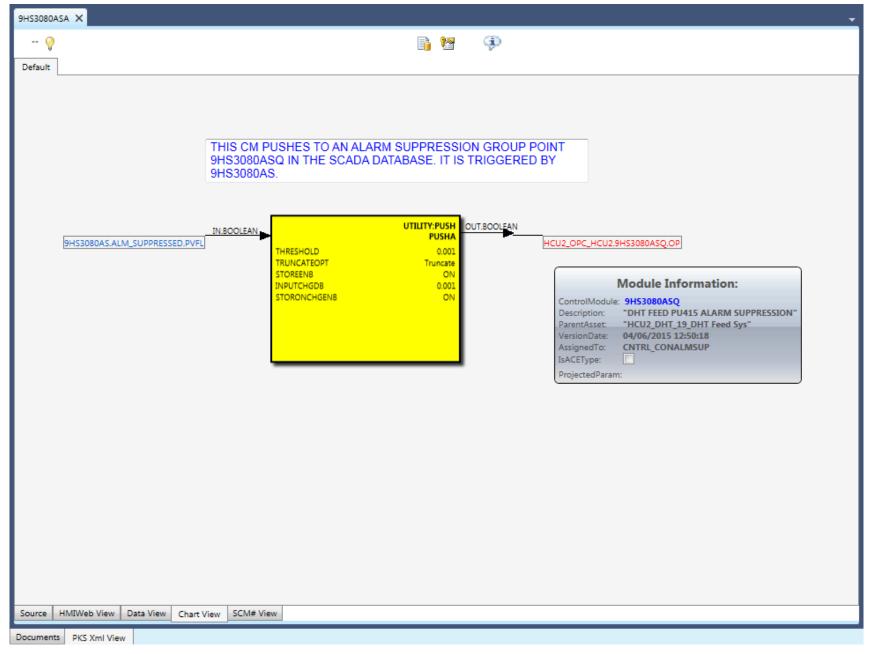
This example shows the conversion of a Hiway Basic Controller Entity named 2FC4272 with its discovered relationships between other Hiway Box slots (entities) and Application Module (AM) entities.



This Control Graph view for the same entity as above shows the Control Strategy and its relationships between LCN Hiway and AM for the focused Hiway Basic Controller Entity 2FC4272.

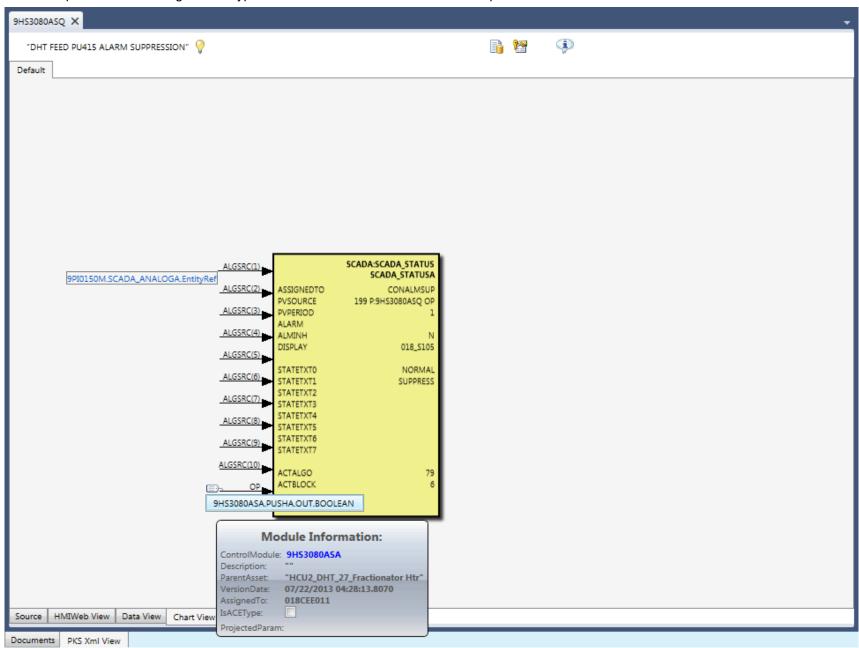


This example shows the relationships between an ACE resident Control Strategy and a SCADA Status point.

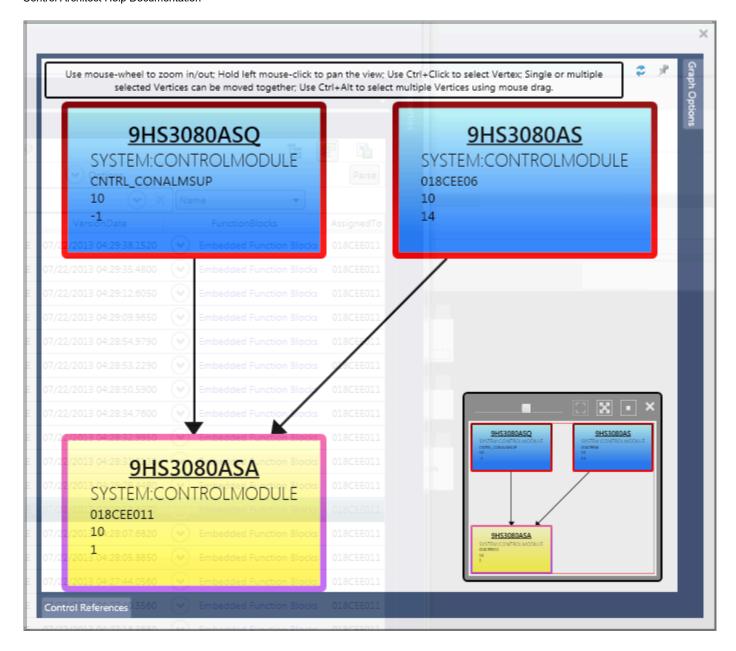


Copyright  $\ @$  2024 by Automation Software Technology, Inc. All Rights Reserved.

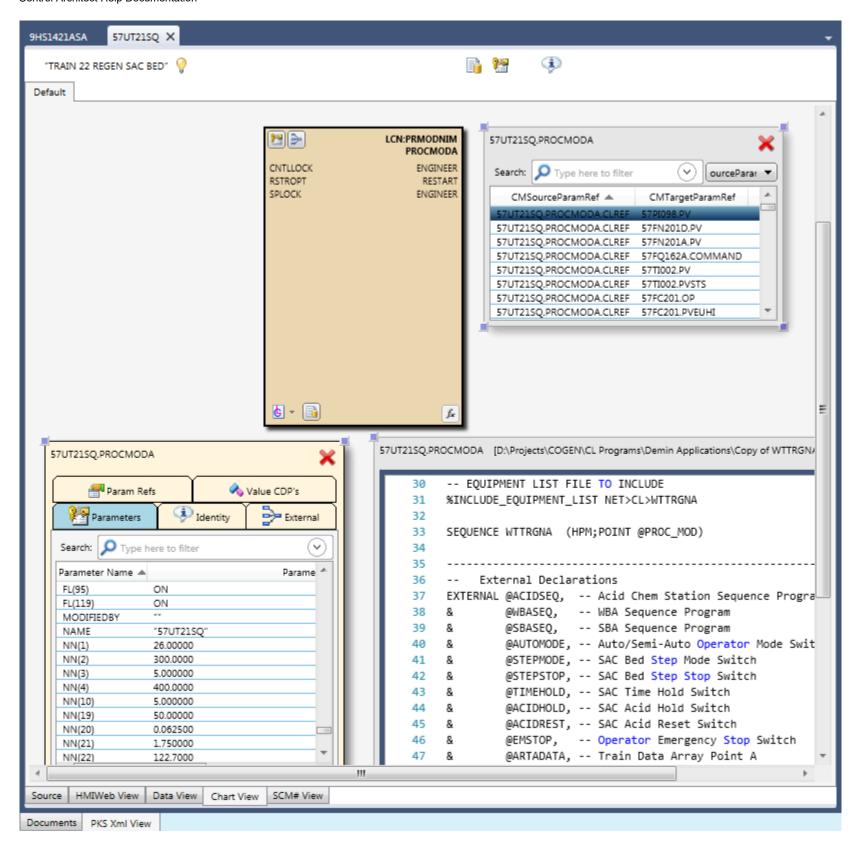
This example shows the navigated to Hyperlink from above for the SCADA Status point.



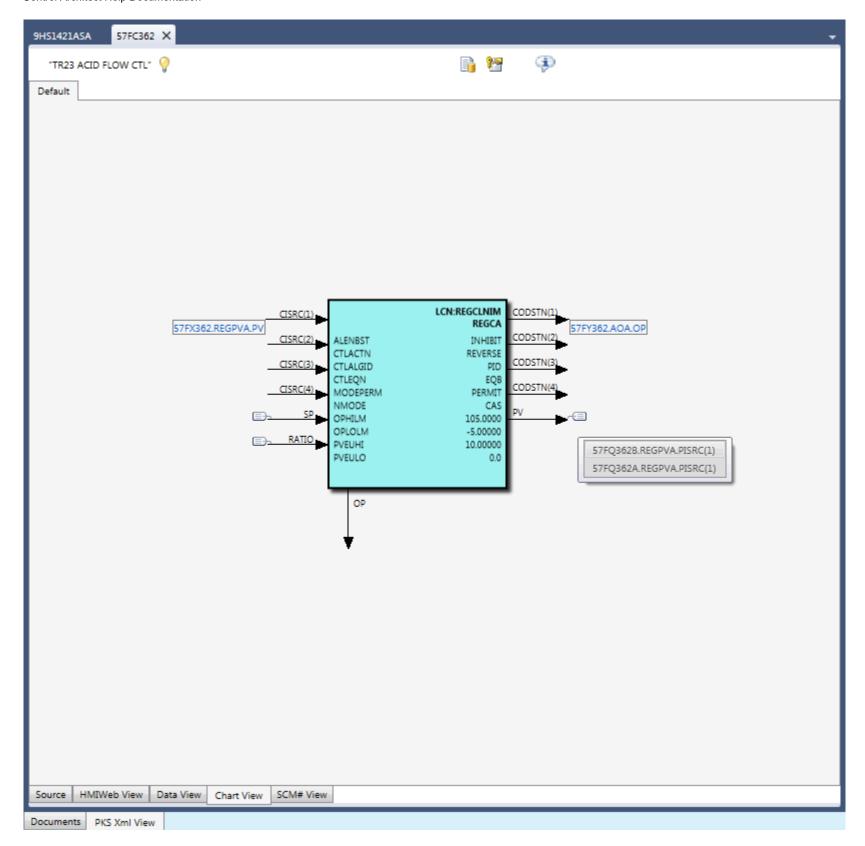
This example shows the Control Graph for the focused Entity 9HS3080ASA that shows the relationships between other CEE resident strategies and the SCADA entity that resides in the SCADA Controller *CONALMSUP*.



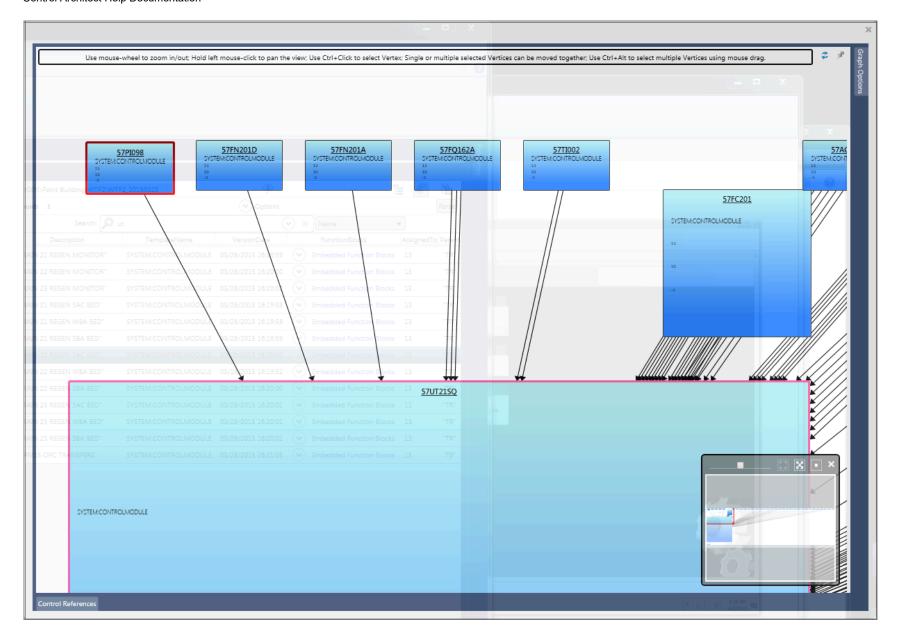
This example shows an XPM Process Module data point. Other views invoked from the Block Adorner show the CL file references, Process Module internal parameter values, and the CL Sequence source code.



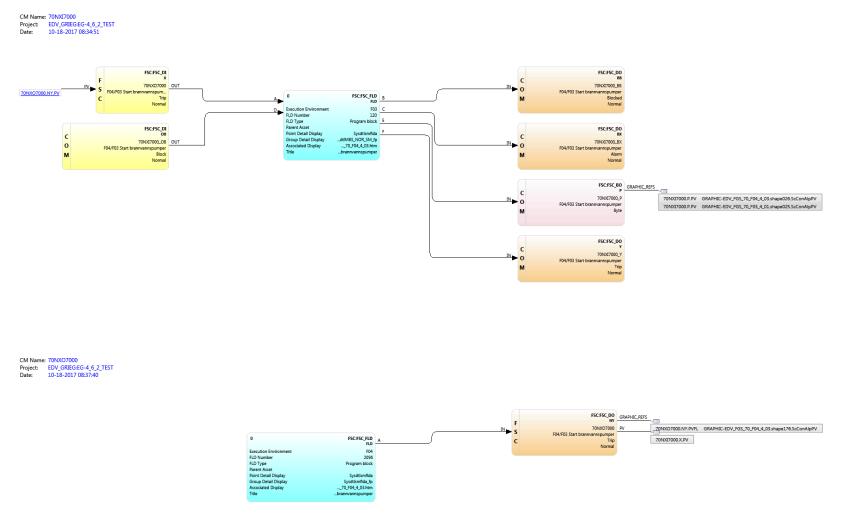
This example shows an XPM Regulatory Control Entity with relationships to other XPM entities.



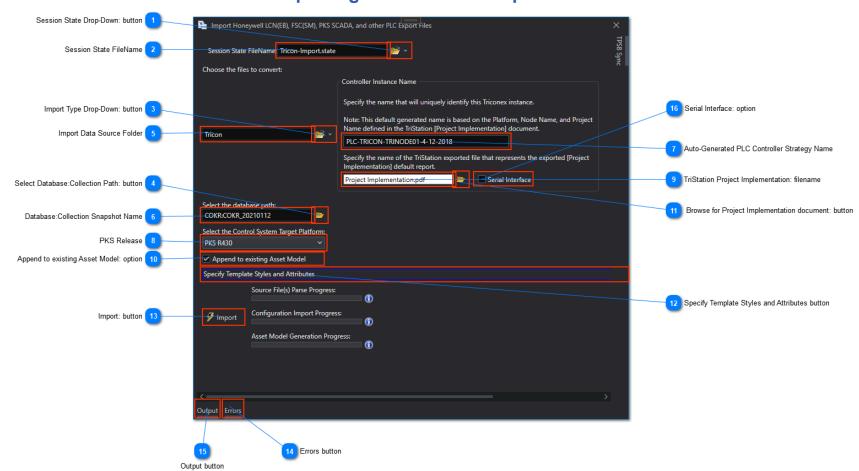
This examples shows the Control Graph for the focused Process Module Entity 57UT21SQ. All references identified in the CL Sequence program are shown as connection relationships to the bound Process Module data point.



Example FSC FLD Control Modules showing FSC Peer reference to another FLD located in another FSC Controller. HMIWeb graphic references are also visible on the FLD blocks.



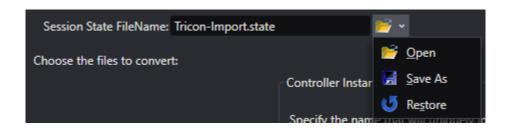
## **Importing Triconex PLC Export Files**



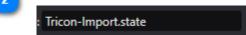
# Session State Drop-Down: button



Click the Drop-Down button to Open, SaveAs, and Restore Session State files.



### Session State FileName

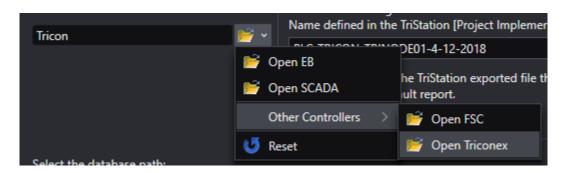


Displays the current selected Session State. Users can choose to save the Import Session properties to a Session State file that can be later restored for quickly importing the same or a similar set of data source files.

# Import Type Drop-Down: button



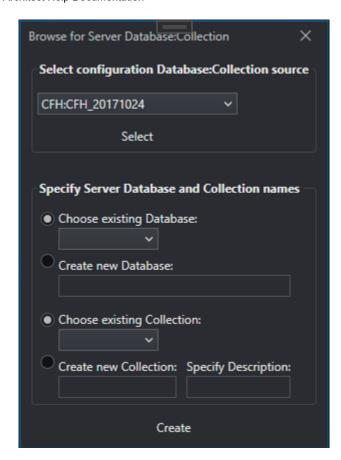
Click this drop-down button to select the type of Import to execute.



### Select Database:Collection Path: button



Click this button to select an existing Database:Collection snapshot or to create a new Database:Collection snapshot to import the data sources to



5

### **Import Data Source Folder**



Displays the File System Folder where the data source files exist to import. This will be an export folder that contains the various documents that were exported using the TriStation application software. The following set of documents are expected in this export folder path:

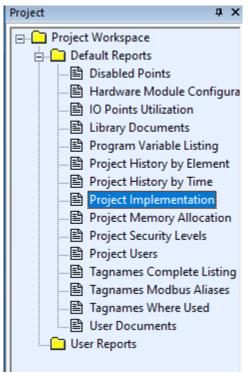
- 1. All Program Logic PDF documents { Note: users can name them in any convenient manner as the import operation will rename them appropriately to the name of the program. }
- 2. "Project Implementation" PDF document
- 3. Program Variable Listing.xls
- 4. Tagnames Where Used.xls
- 5. TagnameDeclarations.xls

Program Logic PDF document files.



Users can choose to name these program logic PDF document files as in 1.pdf, 2.pdf, etc. The import operation will read the Title Block for each Program Logic PDF document and rename them to the program name.

The "Project Implementation" document must be exported as a PDF type so that the Project Information header content is there for reading.



File name: Program Variable Listing.xls

Save as type: Microsoft Excel (97-2003) Data-Only (\*.xls)

Excel (\*.XLS) type data files must be exported using the option

The following documents are required to be exported as Microsoft Excel (\*.xls) Data-Only types:

- 1. Program Variable Listing.xls
- 2. Tagnames Where Used.xls
- 3. TagnameDeclarations.xls

# Database:Collection Snapshot Name



Displays the selected Database: Collection snapshot name.

# Auto-Generated PLC Controller Strategy Name

PLC-TRICON-TRINODE01-4-12-2018

Displays the auto-generated name of the PLC Controller Strategy that will be created during import.

You can choose to manually edit this name, but its uniqueness must be guaranteed as other PLC types can also be imported into the same Collection snapshot.

#### PKS Release



Click this drop-down button to select the PKS Release to use when generating PKS compliant Control Module strategies to contain the PLC import data.

It is recommended to use a mimimum of R430 due to the new Chart View visual rendering model that is now used due to changes that occurred in Control Builder Chart View rendering beginning with PKS Release 430.

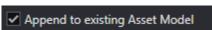
#### **TriStation Project Implementation: filename**

Project Implementation.pdf

Displays the name of the TriStation "Project Implementation.pdf" file that is exported from the TriStation application. This is the default name that is expected. If users choose to use another name, then they can click the browse button to select another file that contains the expected Project Implementation details.

This project detail information is used to extract various data types such as NodeName, ProjectName, DownloadDate, etc. to autogenerate a unique PLC Controller Strategy name.

#### Append to existing Asset Model: option



Choose this option to append data that is imported to the existing Database:Collection snapshot.

If you do not choose to append, then any existing configuration data in the selected Database: Collection snapshot will be deleted prior to importing the new data sources.

If the data being imported has already been appended to this Collection snapshot, then reimporting it will simply delete all document data associated with the unique Triconex node. Existing generated Control Strategies that represent the PLC Controller and Program Logic will also be overwritten during this reimport operation.

#### **Browse for Project Implementation document: button**



Click this button to browse for the PDF document file that contains the TriStation Project Implementation information.

### Specify Template Styles and Attributes button

Specify Template Styles and Attributes

Click this Expander to view the Templates and Styles that are used when generating the Control Module blocks. The TriStation import makes blocks are generic to all PLC type strategies used to represent a PLC Controller node and its Program Logic strategies.



Import: button



Click this button to begin the Import operation.

Errors button



Click this Tab item to view any warnings or errors that occurred while importing the data sources.

Output button



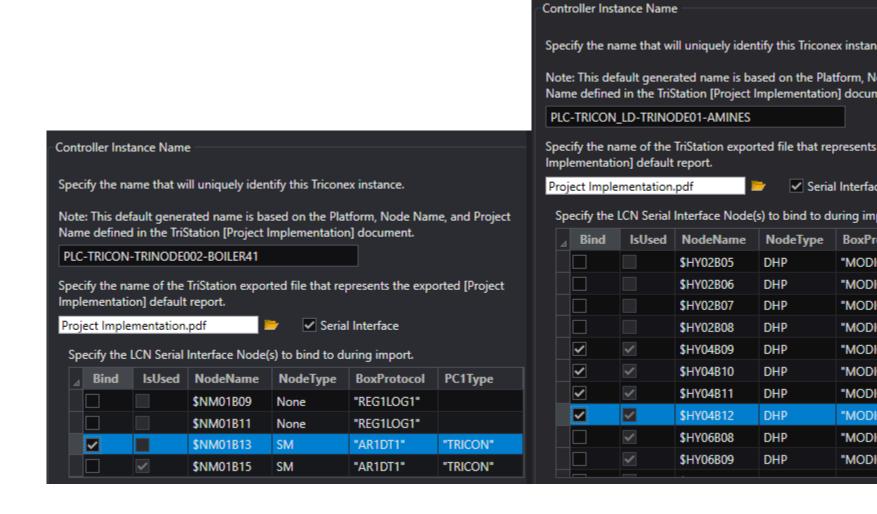
Click this Tab item to view any info and debug information that occurred while importing the data sources.

🥋 Serial Interface: option

Serial Interface

Choose this option to bind the Triconex project data set with an LCN interface type { DHP, PLCG, SMM }

Checking this box will display a list of available LCN nodes read from the selected Database:Collection snapshot as shown below:



# **Getting Started using Triconex Integration**

Control Architect supports the importing of Triconex TriStation exported project configuration data for integration of PLC Modbus data through PKS PCDI Channel blocks and PKS SCADA point blocks. Triconex integration is also supported for all LCN interface types including { DHP, PLCG, SM }. The import process imports exported TriStation documents and generates Control Modules that contain a single embedded block to represent a PLC Controller and blocks to represent PLC Logic Programs. Users can now see the Modbus address and PLC Alias name being referenced in their Control Strategies when viewed in a Chart View. A Program Logic viewer allows users to view the Logic as a PDF view. A Variable list allows users to navigate to its usage in the Program Logic.

#### **Prepare PKS Configuration Snapshot**

A new PKS configuration Collection snapshot must be created using a minimum Control Architect release of 21.11.3. Beginning with this release during the import of PKS Server configuration special processing is added for PCDI Channel Block connections to support PLC Modbus connectivity. Existing configuration snapshots created with an earlier release are not supported for PLC Modbus connectivity.

If your configuration project consists entirely of SCADA, then having to rebuild existing configuration snapshots is not required as there are no additional connection processing required for SCADA points.

#### **Exporting data from TriStation application**

#### **Building the Project Export Folder Content**

The data required for importing into a configuration project Collection snapshot must reside in a single folder in your file system. This folder path is then referenced in the Import tool when choosing to import Triconex type PLC configuration.

#### **Project Implementation Document**

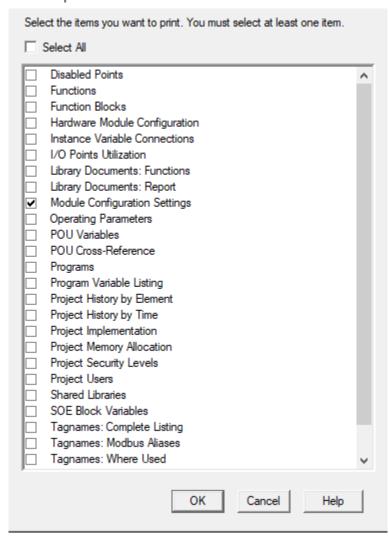
To begin importing TriStation exported data sources, the Project Implementation default report must be displayed and saved as a PDF document type. This document will be read and its Project Information Header Block data parsed for specific properties that includes { Project Version#, Project Description, DownloadedOn date, DownloadedVersion, NodeName, NodeNumber, Platform, TriStation Version#, Last SavedOn date, and the Project Name. These properties are used to generate a unique PLC Controller Strategy name and these property values are cached in this new PLC Controller Strategy to support future version comparison operations.

#### **Module Configuration Settings Document**

The Triconex Node configuration settings document needs to be exported as a PDF document type named "Module Configuration Settings.pdf". This document provides the IP addresses used for both Primary and Secondary that are used to map to an existing PCDI Master or SCADA Channel.

Users can export this document from TriStation by using its File-Print All menu item when the [Controller] tab is selected.

#### Print All Options



#### **Program Logic PDF Documents**

All Program Logic documents must be exported from TriStation as PDF document types using the File-Print All menu item. This will display a "Print All Options" window that allows you to choose only the "Programs" item. While printing to a PDF the user will be prompted to provide a SaveAs PDF filename, which for expediency you can choose to name the programs documents as in 1.pdf, 2.pdf, etc. During import these Program Logic documents will be read and their Title Block data parsed to extract the actual name of the Program Logic that is then used to rename the actual PDF documents prior to importing and to also use as the name of the new generated Program Logic strategies. These Program Logic strategies allow the user to launch a Program Logic viewer window for viewing logic and for navigating its used variable list.

#### Excel (\*.XLS) Data Files

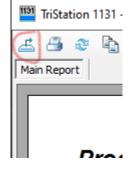
The following Excel documents must be exported from TriStation as Excel \*.XLS Data-Only types.

- Program Variable Listing.xls
- TagnameDeclarations.xls
- Tagnames Where Used.xls

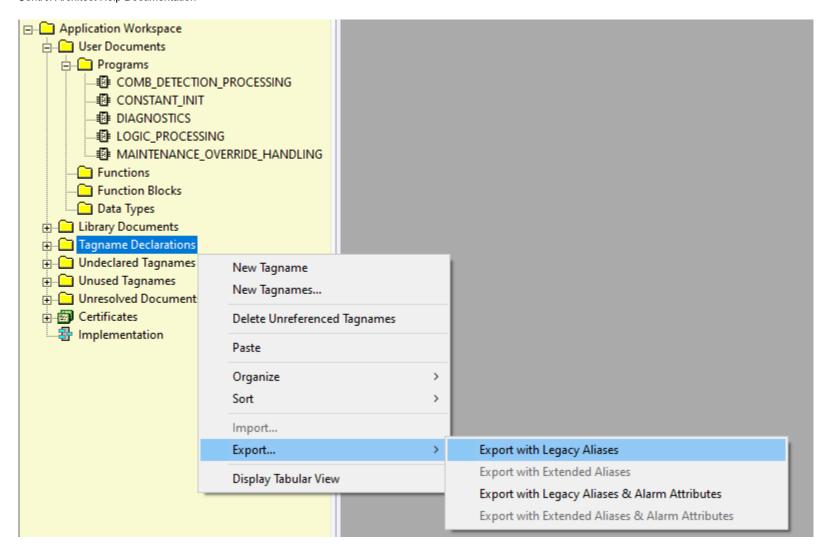
Export these documents from TriStation by:

- 1. Select the Project tab
- 2. Mouse double-click the Program Variable Listing Default Report
- 3. In the report view, click the export button located at the top left of the views Toolbar
- 4. Choose to export as *Microsoft Excel Workbook Data-Only (\*.xls)* type. Please note that CA currently only supports the XLS and not the XLSX document types.
- 5. Save this document to your common data export folder
- 6. Do this for each of the 3 required Excel documents.

The TriStation Export Toolbar button used to export "Program Variable Listing" and "Tagnames Where Used" Excel XLS documents.



The TriStation method for exporting the "Tagname Declarations" Excel XLS document:



#### Importing TriStation exported data into a Control Architect configuration project

#### **Specify the Project Implementation Document**

The "Project Implementation" PDF document must be present in the export folder. It is recommended to use its default name, but if you choose to rename differently, then you can browse for this renamed document file.

If the project implementation document is found, then it is read and its Project information header block is used to generate a unique PLC Controller Control Module name.

#### **Renaming of Program Logic PDF Documents**

As noted above, during import if the name of the Program Logic drawing does not match its identifier as displayed in its Title Block, then the PDF document will be renamed.

### **Modbus Declaration Map**

During import a mapping of all variables defined in the TriStation project is created that also specifies where the Variables are used along with their Modbus Alias name and Modbus Address. This mapped document is used to resolve PCDI Channel and SCADA point block Modbus address references during the import operation. This map is also saved to the generated PLC Controller CM Strategy as a JSON type data collection.

#### Possible Errors when Resolving Modbus Connectivity

It is expected that no errors will occur during import, but reality is that at times what is configured in the DCS won't necessarily match what the PLC project has defined. For example a SCADA point strategy could be referencing a Modbus Address that is no longer defined in the PLC project where the SCADA point should have been either deleted or updated to a new address. Failure to resolve these Modbus connections will be noted as an error in the Error tab view. Errors listed in this view are often indicative of a defect in the configuration.

Modbus references that could not be resolved will be displayed as an invalid Modbus reference.



#### Import results

Upon successful completion of an Import operation documents will have been imported into a MongoDB Database Collection that is logically named "PLCAssets". The documents that were imported consist of the Program Logic PDF files, Modbus Alias mapping, and Excel files.

#### **Control Module Strategies that are created during Import**

During import a number of strategies that represent both the PLC Controller node and its contained PLC logic programs are generated as Control Modules to fit within the PKS Control Strategy schema. Program Logic CM's are contained within the PLC Controller CM. Users can view the assigned and containment relationships by choosing to group the configuration Data View.

A single Control Module is generated to represent the PLC Controller node itself.



Control Modules that represent PLC Program Logic are named using the format: PLC-NodeName-LP-ProgramName where NodeName is the name of the PLC node extracted from the "Project Implementation" document and the ProgramName is the name of the Logic Program. i.e. PLC-TRICON1-LP-COMPRSD.



#### **Overwriting Existing Triconex Import**

Users can choose to overwrite an existing import. If the imported PLC Controller Strategy name is the same as an existing, then it along with all its documents and generated Control Module strategies will be replaced or overwritten. Connections that were generated to support SCADA and PCDI relationships for the PLC Controller being imported are deleted prior to reimporting.

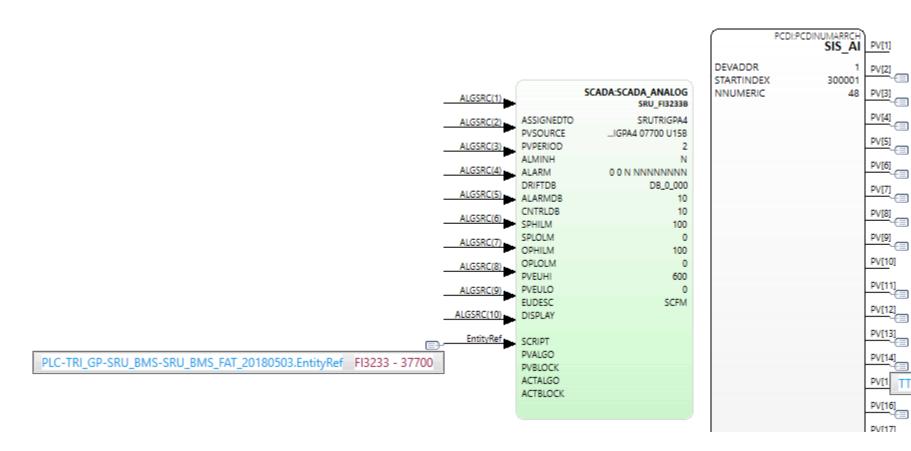
Copyright © 2024 by Automation Software Technology, Inc. All Rights Reserved.

#### **ChartView Visualization**

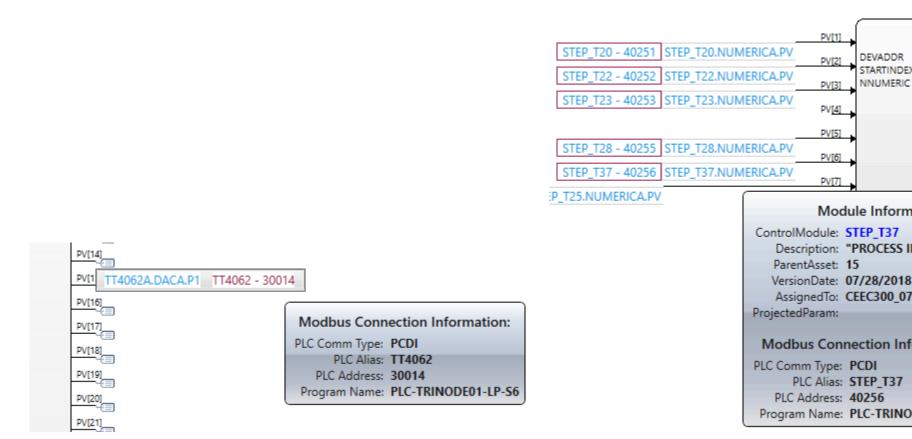
The Chart View has been enhanced to support the visualization of a Modbus connection type. These connection types are expected for PCDI Channel and SCADA point blocks.

#### **Visualization of Modbus Connection types**

Here is an example of what to expect when viewing PCDI Channel blocks or SCADA point blocks within a Chart View.



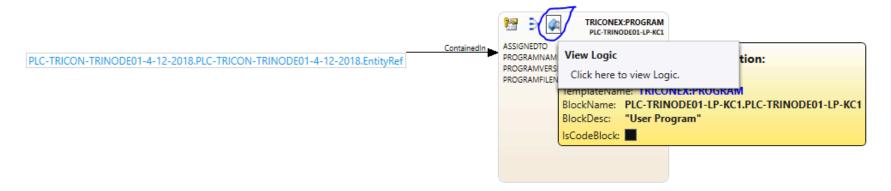
When hovering your mouse over a Modbus connection type, additional information is displayed.



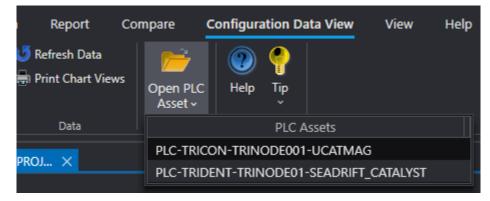
#### **Viewing Program Logic**

The Program Logic associated with a PLC Program Logic block can be viewed by clicking a button that becomes visible in the adorner layer of a Chart View block when the mouse enters a block visual.

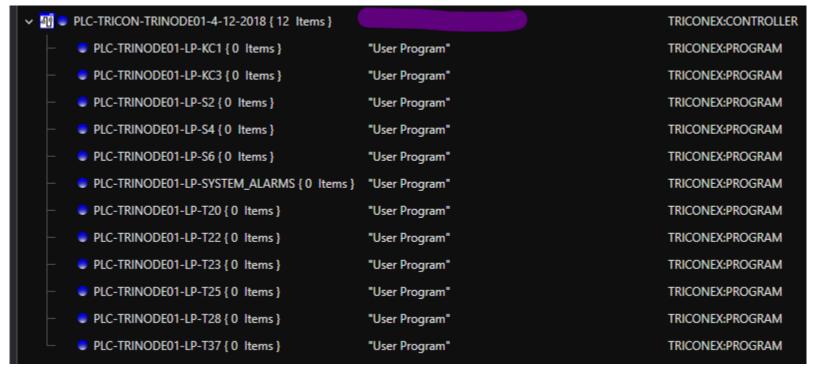
When this *View Logic* button is clicked a <u>Program Logic View</u> window will be launched.



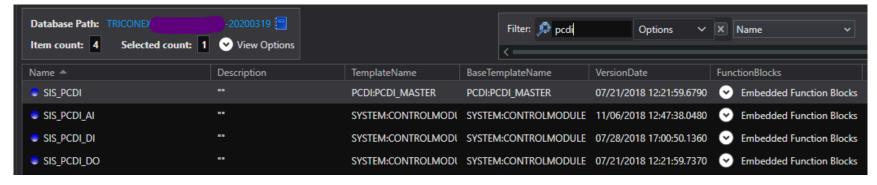
Program Logic can also be viewed from the Data View tab. The Ribbon Toolbar for the Data View tab contains a drop-down button that lists all PLC Assets that are discoverable in the current project configuration snapshot. This viewer allows you to view all logic programs and their variable usage within a single viewer unlike the Program Logic block that is scoped to only that PLC program itself.



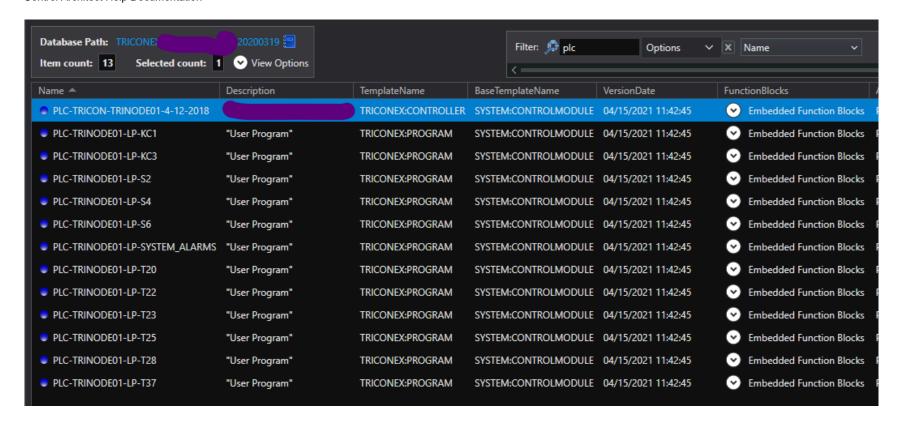
Use the grouping functions to see the containment relationship between a PLC (Triconex) Controller block and its Program Logic blocks. This view is grouped on the "AssignedTo" property.



Users can typically view their PCDI data by using the filter in the Data View to filter on Name or FunctionBlocks with the filter text "PCDI".



Users can also filter on the text "PLC" to achieve a similar related view of strategies associated with a Triconex project import



### Viewing the Manifest Data

The Manifest Data that is created and maintained for each configuration snapshot provides history and information as to the contents of a configuration snapshot. This view is accessible from the configuration <u>Data View</u>. This Manifest View provides the functions that allow you to attach or create a new binding between an existing PLC project import and the current configuration snapshot.

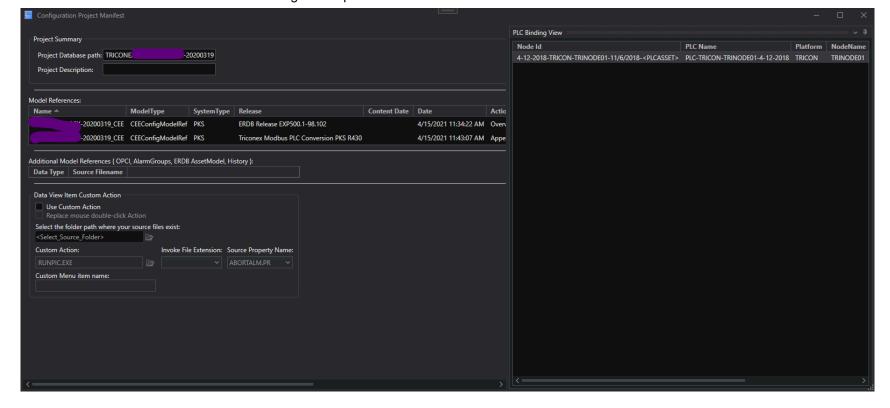
#### PLC Binding to a Configuration Project Database: Collection Snapshot



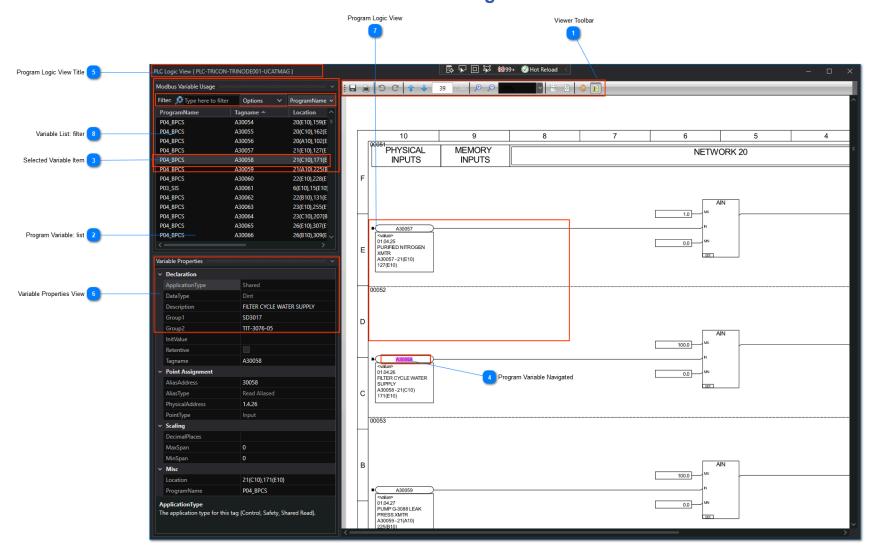
A future release will support the ability to bind an already imported Modbus asset to multiple Database:Collection configuration snapshots.

When a new configuration snapshot is created the user can choose to manually bind an existing PLC project entity to this new configuration snapshot rather than having to reimport the same PLC project information. This assumes that the PLC project configuration itself has not changed.

The existing Manifest View has been enhanced to support a new "PLC Binding View" Tool window. This view provides any future maintenance functions that will become available for working with imported Modbus assets.



### **TRICONEX PLC Logic View**

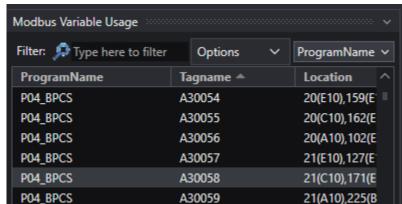


# Viewer Toolbar



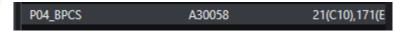
Click the button in this Toolbar to Save, Print, Navigate, Zoom, etc. the view content.

# Program Variable: list



Displays a list of variables that are used in this PLC Program Logic. Mouse double-clicking a list item will navigate to the variable usage in the program and highlight it using the Chart View NonCEE reference defined color. Variable that have multiple usages as defined by multiple sheet locations can be navigated sequentially by repeated mouse double-click actions on the variable item.

#### Selected Variable Item



Program variables used in this program show their Program Name, Variable Declaration Name, and Sheet locations data.

# Program Variable Navigated

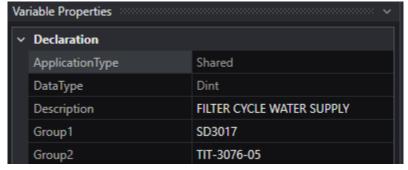
When a program variable in the Variable list is mouse double-clicked, this first items sheet location is brought into view and the found item is highlighted.

# Program Logic View Title

PLC Logic View { PLC-TRICON-TRINODE001-UCATMAG }

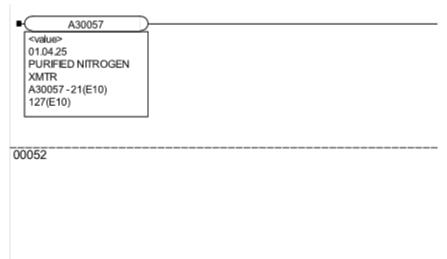
Logic programs are displayed in their own Program Logic View. Views can be identified by the title that is displayed in the Window Title area at the top left of a window.

# Variable Properties View



Displays the properties for the selected Program Variable.

# Program Logic View

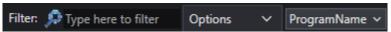


Displays the Program Logic that was exported from the TriStation application and imported into your configuration Collection snapshot as a PDF document type.

The Toolbar provides zoom, page navigation, saving, printing, and window fit functions.

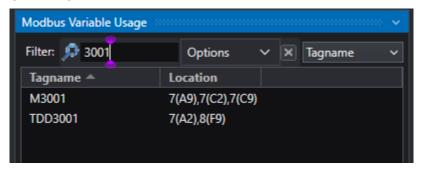
Mouse double-clicking on a Program Variable item in the list will select and bring into view where the Variable is being used. The navigated item is highlighted in a magenta color. Variables that are used in multiple locations in the program logic can be navigated to its next location by again executing a mouse double-click action. If you subsequently double-click to navigate again, but there are no more locations, then the current highlighted program variable has its highlight cleared.

#### 🬎 Variable List: filter

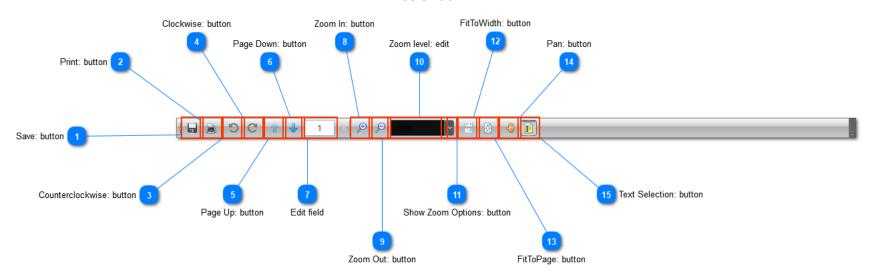


Enter filter text to filter the list of Modbus variables.

The list is now filtered on any Tagname that contains the text phrase "3001". Click the Clear 'x' button to clear the filter text and the filtered list.



#### toolbar



Save: button



Click this button to save the Program logic to a PDF document type.

Print: button



Click this button to print the Program logic.

Counterclockwise: button



Click this button to rotate the Program logic view counter clockwise.

Clockwise: button



Click this button to rotate the Program logic view clockwise.



Click this button to navigate to the previous page.





Click this button to navigate to the next page.

Edit field



Enter a page number to navigate to.

Zoom In: button



Click this button to zoom into the Program logic view.

**Zoom Out: button** 



Click this button to zoom out of the Program logic view.

Zoom level: edit



Enter the zoom level.

11

# **Show Zoom Options: button**



Click the drop-down button to select a zoom level.

FitToWidth: button



Click this button to change the view to fit to the current width.

FitToPage: button



Click this button to change the view to fit to the full page size.

Pan: button



Click this button to pan using the mouse cursor.

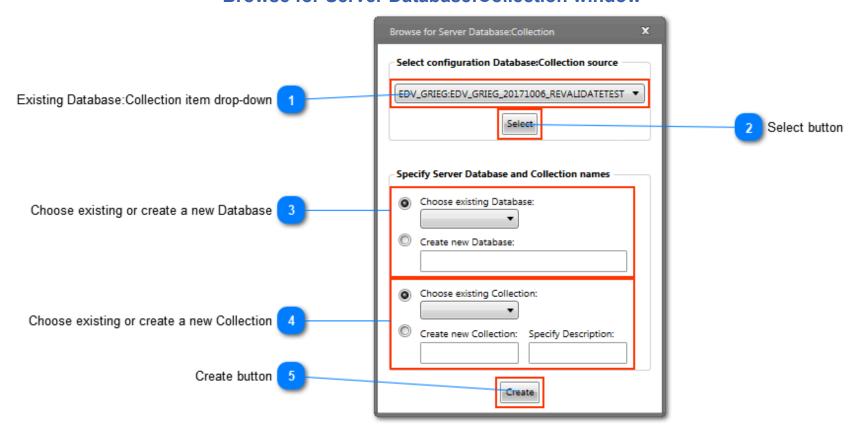
When a program variable item is navigated from the Modbus Variable list, the mode is changed to "Selection".

Text Selection: button



Click this button to select text using the mouse cursor.

#### **Browse for Server Database: Collection window**



# Existing Database:Collection item drop-down

EDV\_GRIEG:EDV\_GRIEG\_20171006\_REVALIDATETEST ▼

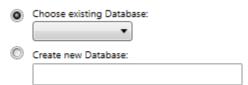
Select an existing Database:Collection from this drop-down list and click the **Select** button.

# Select button



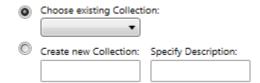
Click this button to select the Database: Collection selection in the drop-down list of existing Database: Collection items.

# Choose existing or create a new Database



If the Database already exists, then select it from the drop-down list of existing Database items. Otherwise, specify a new Database name to create in the Mongo Server.

#### Choose existing or create a new Collection



If the Collection already exists, then select it from the drop-down list of existing Collection items found for the existing Database. Otherwise, specify a new Collection name to create in the selected Database. Alternatively specify a description for this new Database:Collection.

#### Create button



Click this button to create the new Database:Collection in the Mongo Server.

#### **Stats View**

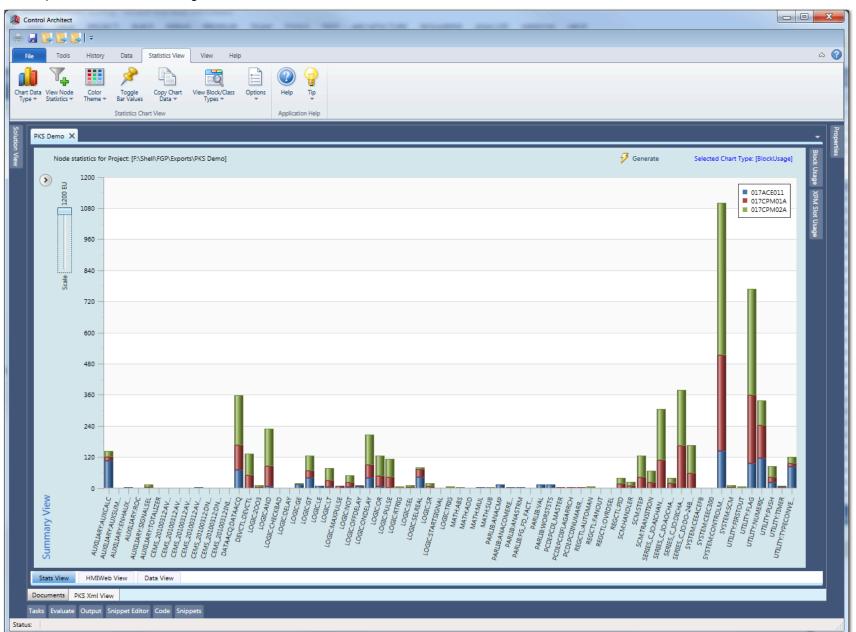
This view is intended for Engineers who require detailed knowledge and understanding of their Process Controller resource utilization to help make better informed decisions on Control Strategy modifications or changes to Process Controller strategies as a result of modernization or new projects.

This view shows graphical Function Block usage stats found in your XML configuration project. Both PKS and LCN type configuration projects are supported. Block instances, Processor node Memory Unit (MU), and Processing Resource Unit (PU) calculations are visually shown in a bar type chart.

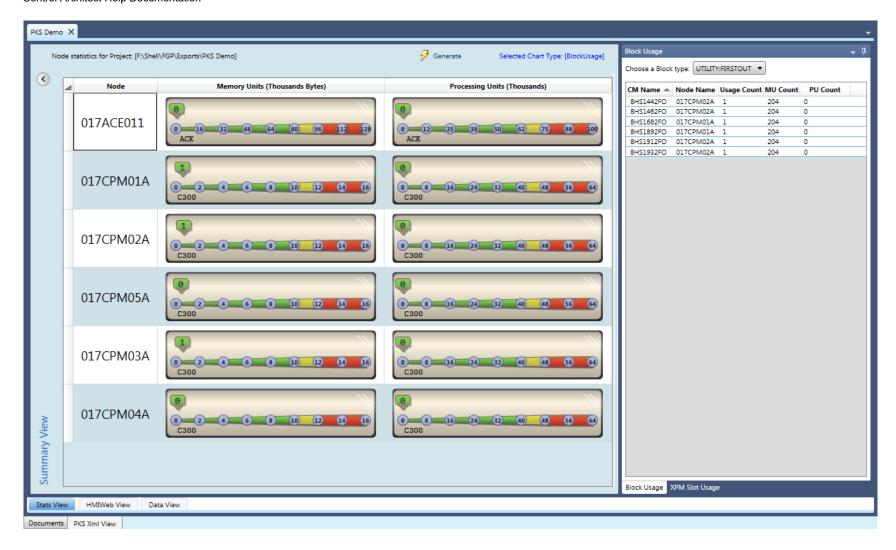
For LCN XPM type projects this view also provides additional PMM Control Processor point type usage views to visually identify used and spare slot ranges.

A hardware view for the Honeywell legacy LCN Control System also provides a visual network view of the LCN system including ENIM's and EHPM's.

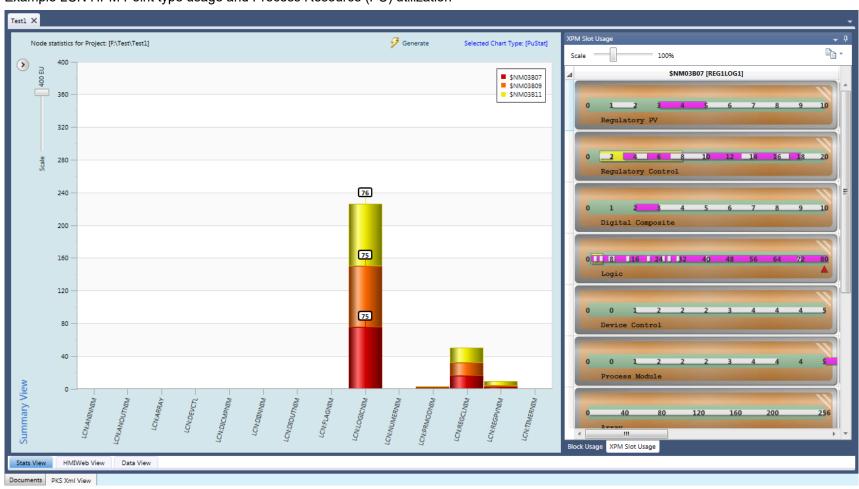
#### Example PKS C300 Block usage

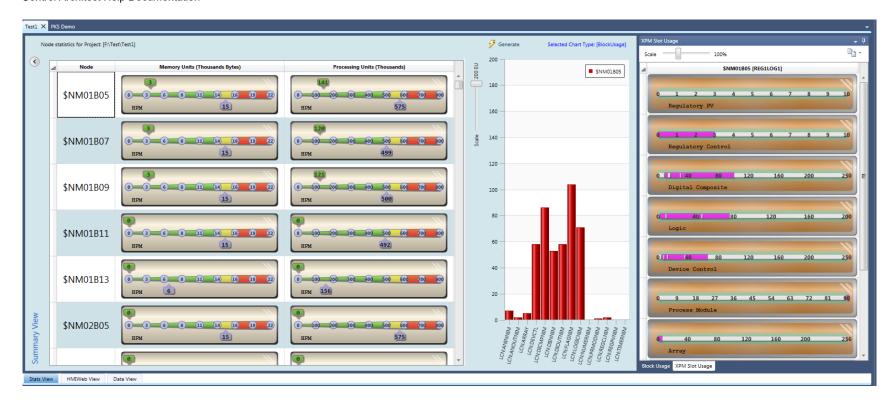


Example PKS C300 Summary and detailed Block usage

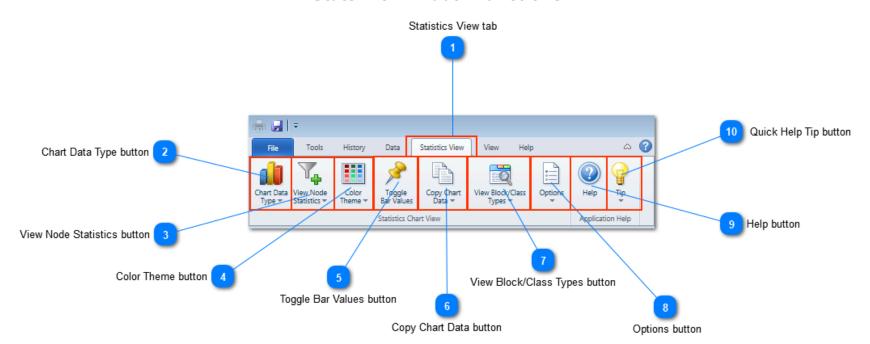


#### Example LCN HPM Point type usage and Process Resource (PU) utilization





#### **Stats View Ribbon functions**



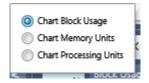
# Statistics View tab Statistics View

Select this tab to execute functions on the active Stats View.

# , Chart Data Type button



Click the drop-down arrow to access the data type to show in the Bar chart. Node Block usage data is shown as a Bar type chart visual. You can choose multiple Controller nodes to view in the chart as a set of stacked Bars where the bar color uniquely identifies a specific Controller node name.



- Choose the Chart Block Usage option to display a bar chart of block instances used for the selected Controller node.
- Choose the **Chart Memory Units** option to display a bar chart of Memory units consumed for each Function Block type for the selected Controller node.
- Choose the **Chart Processing Units** option to display a bar chart of Processing Resource Units consumed for each Function Block type for the selected Controller node.

a.

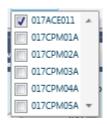


PU calculation for PKS CEE type controllers is not yet supported as Block type PU utilization values are not yet available from Honeywell. An update will be provided when Honeywell provides this missing piece of information.

#### View Node Statistics button



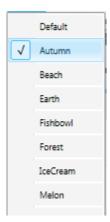
Click the drop-down arrow to select a single or multiple Controller nodes to view its usage data in the Bar chart.



# **Color Theme button**



Click the drop-down arrow to select a color palette to apply to the data bars in the Bar chart. When multiple Controller nodes are selected for display, a color from this palette kind will automatically be chosen to uniquely identify the Controller nodes.



# Toggle Bar Values button



Click this button to toggle the Bar chart value label visibility. You can quickly see the values for each data bar in the Bar chart by toggling their visibility on. Click this button again to hide the data values.

# Copy Chart Data button



Click the drop-down arrow to choose the Copy/Paste option. You can copy data and the Bar chart image to the Windows Clipboard for pasting into Excel. To paste the generated PNG image into Excel, use the Excel **Paste-Paste Special...** Toolbar function and choose the Bitmap type to insert the image from the Windows Clipboard into the active Worksheet.



- Choose the Copy Image and Data option to copy both the chart raw data and an image of the Bar chart to the Windows Clipboard.
- Choose the Copy Image Only option to copy an image of the Bar chart to the Windows Clipboard.

#### View Block/Class Types button



Click the drop-down arrow to view/edit Block and Class type Function Block data that is used in generating MU & PU calculations.



- Choose the View PKS Block types menu option to view/edit the MU & PU values associated with Function Block types.
- Choose the **View PKS Class types** menu option to view/edit the first instance Memory Units assigned to the first instanced usage of a particular Class type.

# Options button



Click the drop-down arrow to choose options to how certain data types are visualized in the Bar chart.



- Choose the **Show FieldBus Devices** option to show each FieldBus device type as a Function Block type. If this option is not checked, then FieldBus devices are not shown in the Bar chart.
- Choose the **Combine CAB and CD Block types** option to show all CAB and CD Block types as a single Function Block type named "CABLOCK" and "CDLOCK".





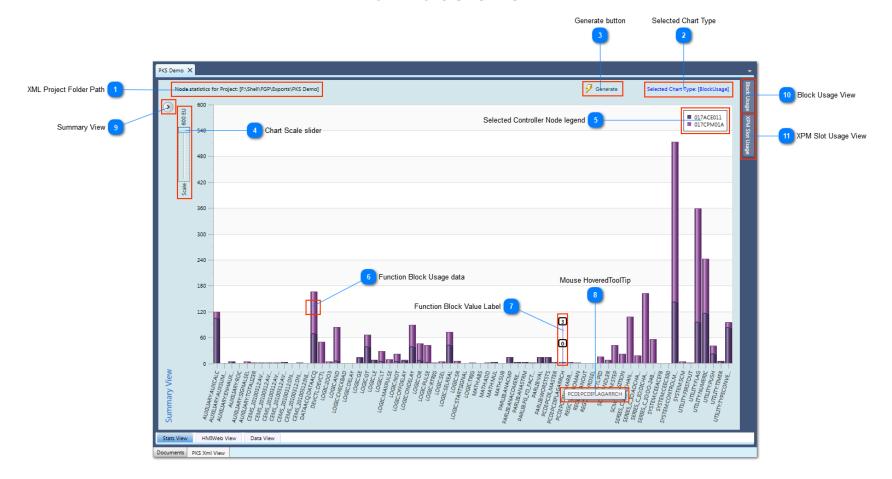
Click this button to view this application help topic.

Quick Help Tip button



Click the drop-down arrow to view quick help tips for working with this Stats View.

#### **Bar Chart Overview**



# XML Project Folder Path

Node statistics for Project: [F:\Shell\FGP\Exports\PKS Demo]

Displays the XML Configuration Project folder path where the Configuration Asset Model resides that is used to generate data visualized in this Bar chart.

# Selected Chart Type

Selected Chart Type: [BlockUsage]

Displays the Chart Type option that was chosen for visualizing data in the Bar chart.

# Generate button



Click this button to generate data for visualization in the Bar chart.

When the data generation has completed, by default Block Usage data is chosen as the Chart Type for visualization in the Bar chart.

#### 🦱 Chart Scale slider



Change the scaling in Engineering units for data presented in the Bar chart. When some data set values are very large and others small you may want to scale down the chart to visually view data bars that have low values.

#### Selected Controller Node legend



Legend displays the selected Controller nodes and the palette color automatically selected to uniquely identify its bar or bar part in the chart.

# Function Block Usage data



This example shows the block usage count for DataAcq type Function blocks associated with Controller Node 017CPM01A.

Copyright © 2024 by Automation Software Technology, Inc. All Rights Reserved.



Shows the value labels associated with a Function block data bar in the chart. These value labels are automatically displayed when the mouse is hovered over the bar or X axis label in the chart. You can also toggle value label visibility for all data bars in the chart using the <u>Toggle Bar Values</u> Toolbar function.

# 8 Mouse HoveredToolTip



A ToolTip is displayed when your mouse is hovered over an X axis label Function Block label. If the label value is too long to display fully in the X axis, its label will be shortened as indicated by the trailing ... characters. Use your mouse to hover over the label to see its full name.

9 Summary View



Click this button expand/collapse the Summary View.

Block Usage View



Hover over this tab with your mouse to show this Block Usage view. You can dock this view when visible by clicking its pin image located in the views top right corner.

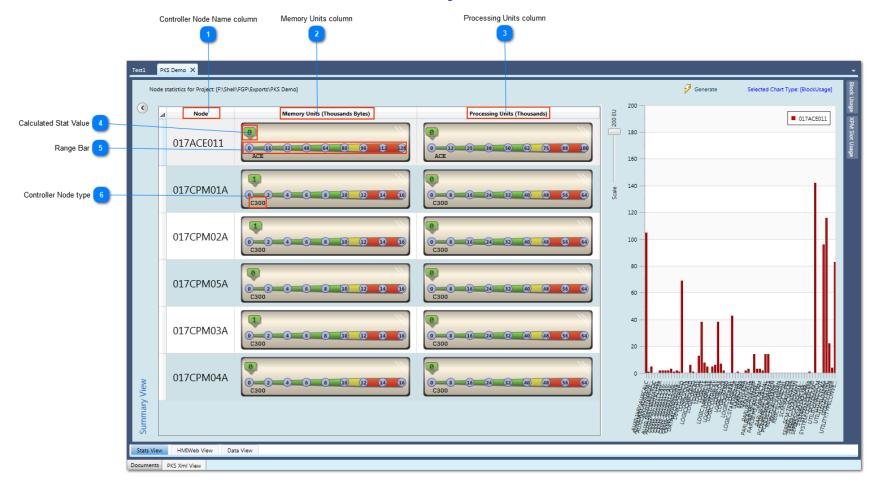
XPM Slot Usage View



Hover over this tab with your mouse to show this XPM Slot Usage view. You can dock this view when visible by clicking its pin image located in the views top right corner.

This view will only contain data for LCN XPM type Controller nodes that exist in the XML configuration project.

### **Summary View**



# Controller Node Name column

Node

Shows the name of the Process Controller Node.

# Memory Units column

Memory Units (Thousands Bytes)

Shows the calculated Memory Units for each Process Control Node.

### Processing Units column

Processing Units (Thousands)

Shows the calculated Process Resource Units for each Process Control Node.



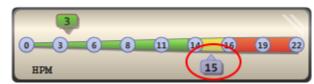
PU calculation for PKS CEE type controllers is not yet supported as Block type PU utilization values are not yet available from Honeywell. An update will be provided when Honeywell provides this missing piece of information.

#### Calculated Stat Value



Indicates the calculated unit value for the resource type. Hover your mouse over this marker shape to see the precise value as a ToolTip.

The example XPM type gauge below shows an additional Marker shape that indicates the allocated MU and PU value for the node. The allocated values are based on the maximum number of slots configured for the specific point types. This additional marker combined with the actual analyzed usage value marker shows clearly how much of the allocated Control Processor memory and processing units are available to use.



# Range Bar



The Range bar shows the maximum range of the units and is divided into three linear ranges shown by the bar colors green, yellow, and red.

The yellow range is configured for 60-75% of range. The red range is configured for 75-100% range. These range colors provide a visual cue of the resource utilization of the processor when looking at the linear gauge that represents a static analysis of the resource load of the Controller node.

The maximum range value is determined by the Controller type. ACE Control processors for example support a maximum of 128K Memory Units. A Memory Unit in a CEE type controller is defined as 1024 bytes = 1 MU.

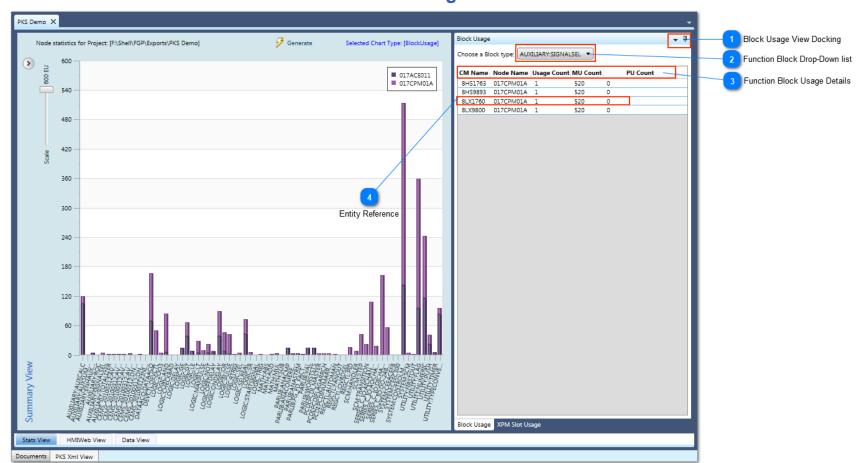


### **Controller Node type**



Shows the Controller type such as ACE, C300, HPM.

# **Block Usage View**



# Block Usage View Docking

You can choose to Dock/UnDock this view by clicking the pin button. You can also click the arrow button to float this view and re-dock the view to another location such as top, bottom, or left sides of the main Stats View window.

# Function Block Drop-Down list

AUXILIARY:SIGNALSEL ▼

Select a Function Block type from this list to view its usage data. This Block list is generated based on the blocks that were found while analyzing your configuration project.

# Function Block Usage Details

CM Name Node Name Usage Count MU Count PU Count

The data presented in this list includes the Entity name (CM Name), its assigned Controller Node name, the instance count found for this Entity, and its calculated Memory Unit (MU) and Process Unit (PU). For PKS type Control Modules there can multiple instances of this Block type in the same CM, whereas LCN types can only contain a single instance.

# Entity Reference

8LX1760 017CPM01A 1 520 0

You can left mouse double-click on an Entity item in this list to view its Chart View rendering.

You can also right mouse-click on an Entity item in this list to show its context menu and select the menu item to view its Chart View rendering.

Open Chart View

# **XPM Slot Usage View**



# XPM Slot Usage View Docking



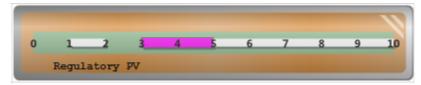
You can choose to Dock/UnDock this view by clicking the pin button. You can also click the arrow button to float this view and re-dock the view to another location such as top, bottom, or left sides of the main Stats View window.

# Controller Node Name

\$NM07B33 [REG1LOG1]

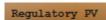
Displays the Controller Node name and its configured ScanRate parameter value for the Point type data visualized in the DataGrid column set of gauges.

# Point Type Range Usage



This gauge shows how slots are used and allocated for the particular Point type.

# Point Type Name



Shows the Point type that is visualized in this gauge.

# **Gauge Scale**

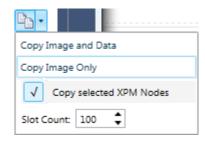


You can scale up or down the gauge visuals.

# Copy/Paste to Excel



You can copy the data visualized in this set of linear gauges to the Windows Clipboard for pasting into  $\underline{\mathsf{Excel}}$ .



- Choose the menu option **Copy Image and Data** to copy both the Controller node data and an image of the linear Point type gauges to the Windows Clipboard.
- Choose the menu option Copy Image Only to copy only the image of the linear Point type gauges to the Windows Clipboard.
- Check the option **Copy selected XPM Nodes** to copy only the Controller node data that has been selected. Leave this option unchecked to copy data for all Controller nodes in the XML configuration project that has been analyzed.
- Specify the number of Point type slots that will be shown in the Excel data. This determines how many slots will be shown for each Point type and the Entity that is assigned to this Slot.

# Maximum Allocated Slot Value



Indicates the maximum allocated range for the Point type.

# Configured Used Range



The magenta linear bar shows the analyzed Used range for this Point type.

# Spare Slot Range



The white linear bars show the analyzed Spare Slot ranges for this Point type. The gauge is capable of showing a maximum of 10 Spare Slot range indicators. If the number of Spare Slot ranges exceeds 10, then a <u>red triangle shape indicator</u> is displayed.

#### Fast Slot Range



The yellow linear bar shows the configured number of fast slots for this Point type.

#### **Number of Visual Spare Ranges (10) Exceeded**

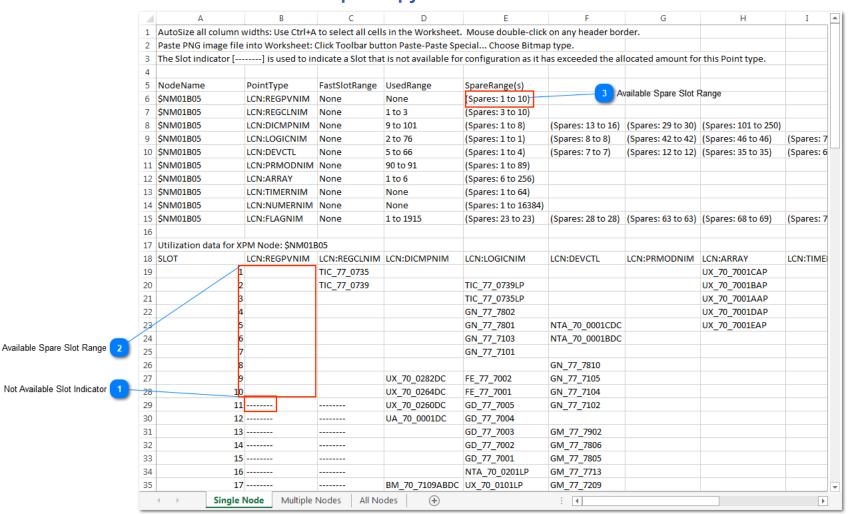


This red triangle shape indicator located near the bottom of the bar at its maximum range indicates that the number of analyzed spare ranges has exceeded the maximum number of 10 Spare Slot ranges that can be visually displayed on this gauge. To view all available spare ranges for this Point type, <u>copy</u> the data to the Windows Clipboard and paste into Excel.



The Point Type visual above also shows a yellow triangle shape indicator located near the top maximum range of the bar that indicates that the shown maximum range value is not true, but has been scaled down to show the used range. Hover over this triangle shape indicator with your mouse to view the true allocated Point type range.

# **Example Copy/Paste to Excel**



# Not Available Slot Indicator

Indicates that this Slot is not available for configuration as the maximum number of slots allocated for this particular Point type = 10.

### 🥱 Available Spare Slot Range



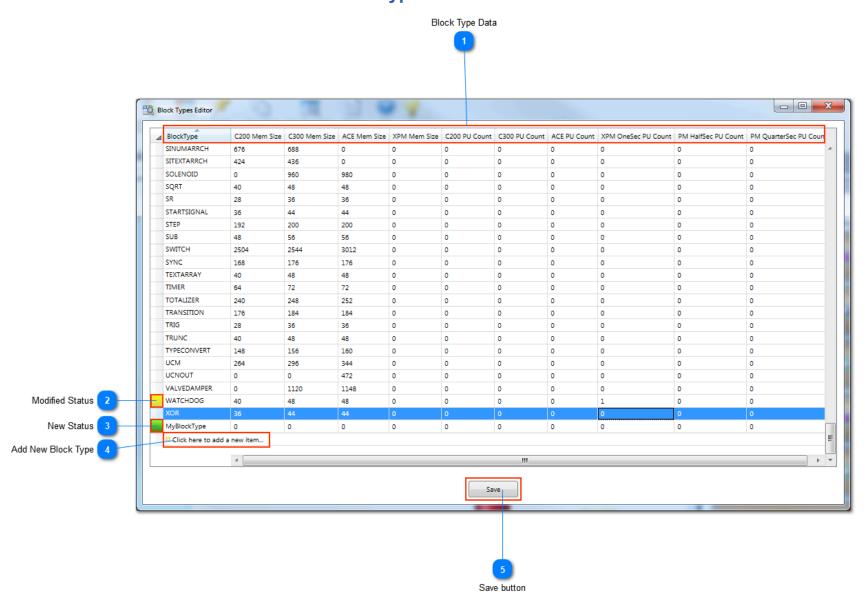
Indicates that slots 1-10 are available for configuration for this particular Point type.

#### Available Spare Slot Range

(Spares: 1 to 10)

The SpareRanges(s) data also indicates the Spare Slot ranges that are available for configuration.

# **Block Types Editor View**



Block Type Data



Shows the Block Type data that is available for calculating Memory Unit and Process Resource Unit utilization stats. All data shown for Memory as bytes. A CEE type Memory Unit is defined as 1024 bytes/MU. LCN Memory units are defined in bytes where 1 byte = 1 MU.

Modified Status



This row header indicator shows that this Block type has been modified.

New Status

This row header indicator shows that this Block type has been added as new.

Add New Block Type

Click here to add a new item...

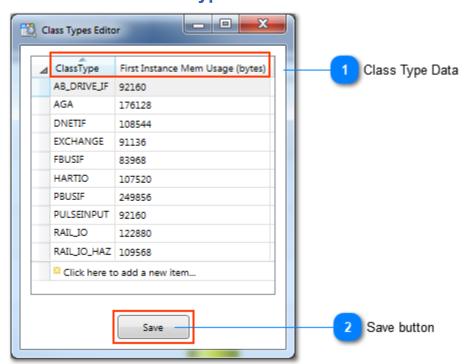
Click here to add a new Block Type row.

Save button

Save

Click this button to save all changes made to the Block types.

# **Class Types Editor View**



# Class Type Data



Shows the Class Type data that is available for calculating Memory units for first instance usage. All data shown for Memory Unit sizes are listed as bytes. A CEE type Memory Unit is defined as 1024 bytes/MU.

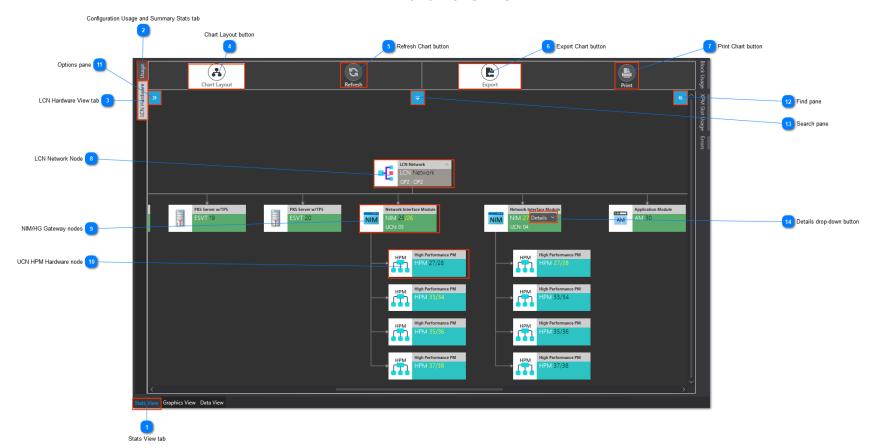
The first usage of a Block type from this class will consume an additional memory overhead as defined in this list.

# Save button



Click this button to save all changes made to the Class types.

#### **LCN Hardware View**



# Stats View tab



Select this tab to view hardware/software statistics for the configuration project that includes template summary usage, hardware usage, etc.

# Configuration Usage and Summary Stats tab



Click this tab within the Stats View to view usage and summaries of configuration templates and hardware.

#### LCN Hardware View tab



Click this tab to view the Honeywell LCN Network view of their legacy Control System.

### Chart Layout button



Click this button to layout the chart diagram in a different tree style.

#### Refresh Chart button



Click this button to refresh and redraw the chart diagram. If a user moves around the contents within the diagram they can click this button to redraw the network layout.

#### Export Chart button



Click this button to export the Chart diagram to either a PNG or BMP graphic file.



#### **Print Chart button**



Click this button to print the Chart diagram to a print device.

8

#### **LCN Network Node**

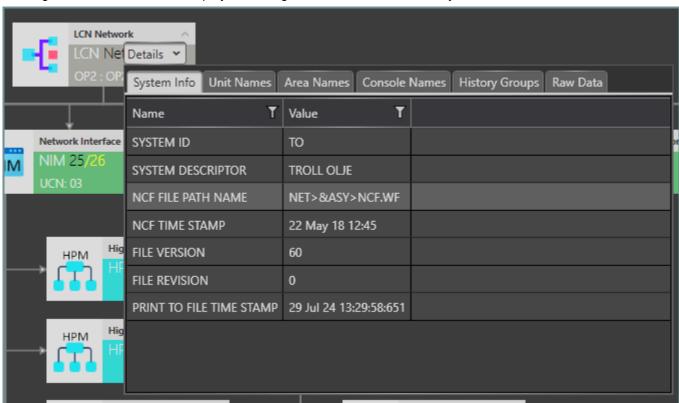


This LCN node represents the root or the actual LCN Network itself. Hover your mouse over this node shape to access a drop-down button that when clicked will show additional detail information associated with the LCN network.

Users must export 2 reports from the LCN Native window to auto-generate this LCN Network diagram. These 2 report files are then imported into Control Architect to auto-generate this diagram.

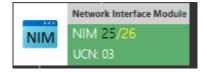
- 1. The LVRLOG.RE report
- 2. The NCF.XX report

Clicking the **Details** button will display the configuration details for the Honeywell LCN Network.

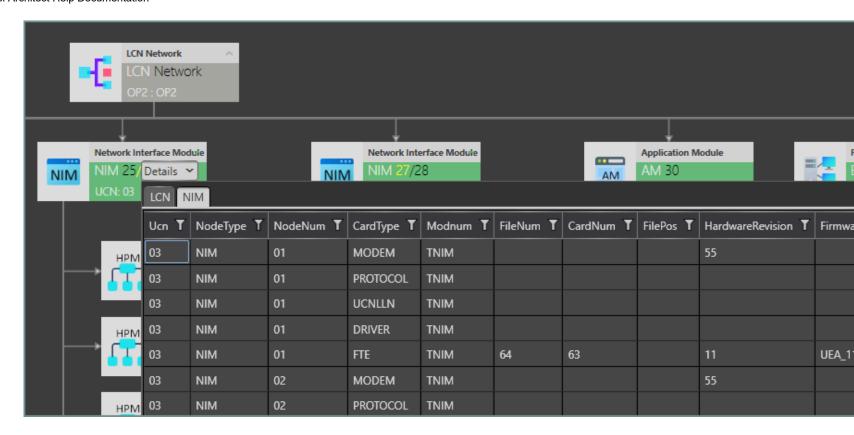




#### **NIM/HG Gateway nodes**



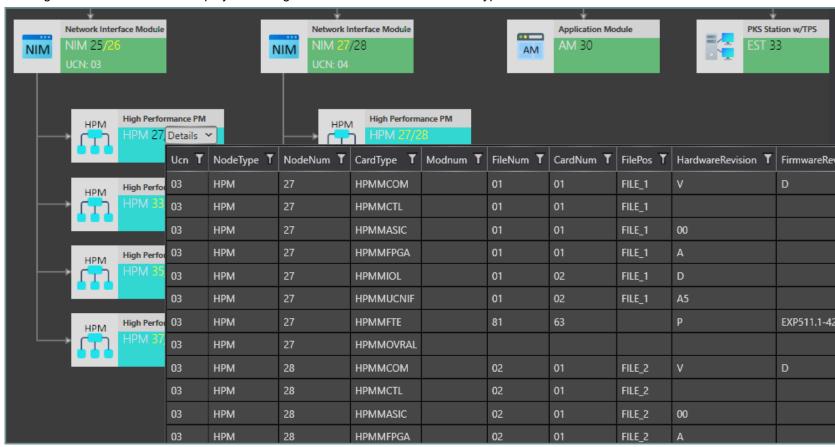
Clicking the **Details** button will display the configuration details for the NIM or HG LCN nodes.



UCN HPM Hardware node



Clicking the **Details** button will display the configuration details for the LCN xPM type nodes.

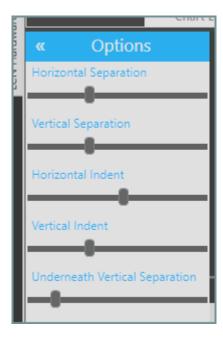


# Options pane

**>>** 

Click this button to expand/collapse the *Options* view.

These options allow you to provide some input as to how shapes are laid out.



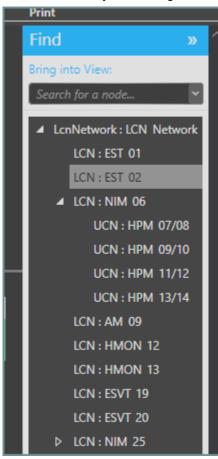
12

#### **Find pane**



Click this button to expand/collapse the Find view.

This view allows you to navigate the LCN node structure from a Tree like view selection or searching in a drop-down box.



13

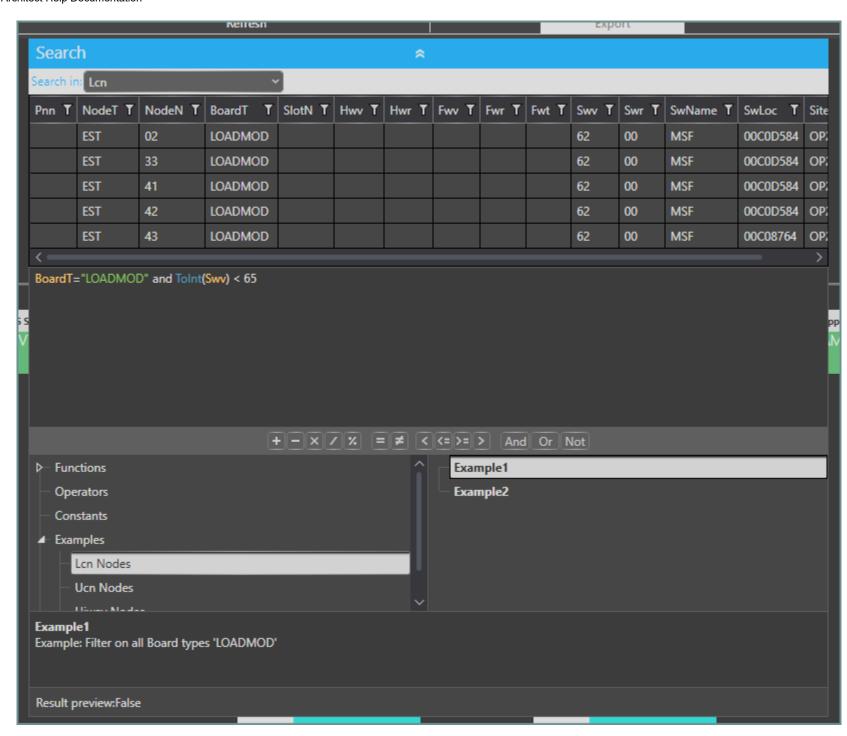
#### Search pane



Click this button to expand/collapse the Search view.

This view allows you to search and filter the detail data of the shape nodes for particulate details. This expression view support user expressions to filter the data view.

Users must choose between searching in LCN, UCN, or HWY type data nodes to initiate a search.



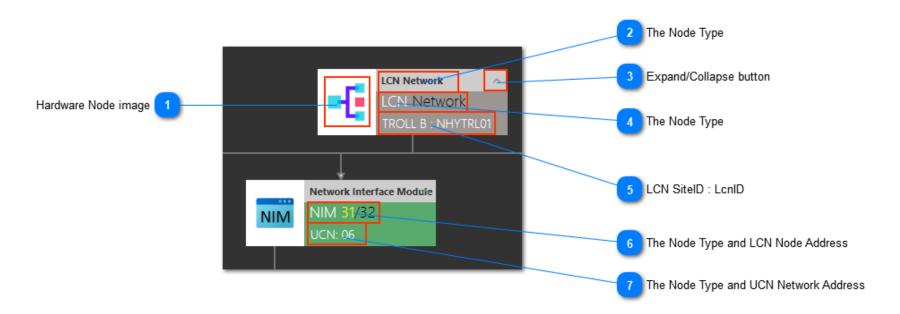
# Details drop-down button

Details ~

Each node shape when hovered with the mouse will display this Details drop-down button. When clicked it will show the respective configura



#### **Visual Structure**



Hardware Node image



Displays an image that represents the type of hardware node.

The Node Type

LCN Network

Displays text that indicates the type of Hardware node this shape represents.

Expand/Collapse button



Click this button to expand/collapse child nodes to this node.

The Node Type

LCN Network

Displays text that indicates the type of Hardware node this shape represents.

LCN SiteID : LcnID

TROLL B: NHYTRL01

Displays the LCN Site and LCN Id's extracted from the LVRLOG.RE LCN report file.

The Node Type and LCN Node Address

NIM 31/32

Displays the type of LCN Node and its Hardware Address on the LCN.

The Node Type and UCN Network Address

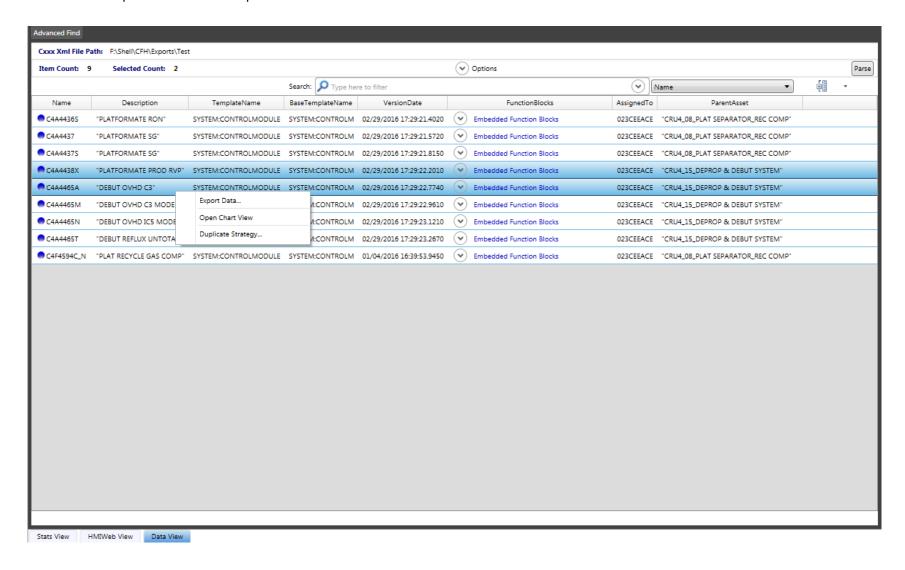
Displays the type of LCN Node and it UCN Network Address.



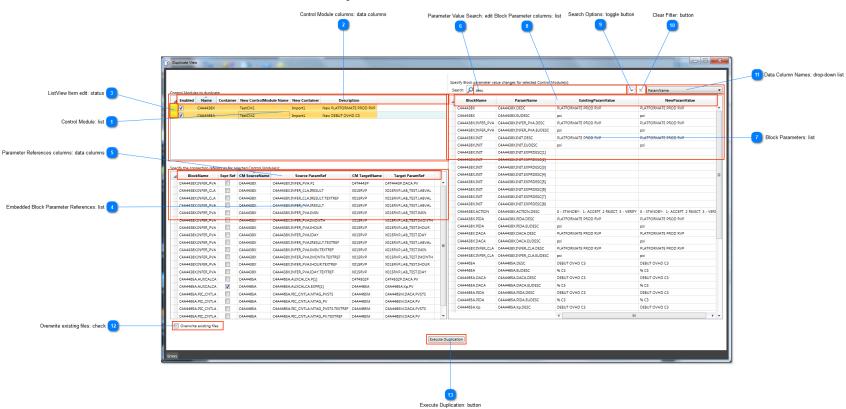
## **CM Duplicate View**

This view and function allows users to quickly duplicate selected Control Module item(s) in the Data View. This function is similar to PKS Control Builder Copy function, but provides a more complete function set:

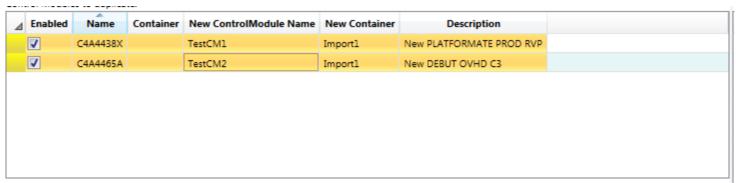
- Control of how Parameter values are written
- Containment options for the new duplicated CM's



## **Duplicate View window**



## Control Module: list



The set of Control Module item(s) that were selected from the Data View for duplication.

To enable duplication users must:

- 1. Provide a new Control Module name
- 2. Optionally specify a new Description for the duplicated CM
- 3. Optionally specify a new Container CM name for the duplicated CM

When you select a CM item in this list, all Parameter references for its embedded Function blocks are added to the list below it. The *Target ParamRef* data column is editable. The Parameters found in the embedded Function blocks are also added to the list at the right of the view.

## Control Module columns: data columns



- Enabled: Indicates if this CM item is enabled for duplication. You cannot enable a CM item for duplication until you have specified a new name for the duplicated CM.
- Name: The name of the Control Module that will be the basis for the duplication
- Container: The existing Container Module if this CM is currently embedded
- New ControlModule Name: Specify a new name for the duplicated Control Module
- New Container: Optionally specify a new Container name if the duplicated CM is to be re-parented to a new containing Control Module. The Container name specified here must already exist in order to reference it.
- Description: Specify a new description for the duplicated CM

### 🔁 ListView Item edit: status



Indicates the current edit status for a ListView item.

White: UnchangedGreen: AddedYellow: Modified

## **Embedded Block Parameter References: list**

⊿	BlockName	Expr Ref	CM SourceName	Source ParamRef	CM TargetName	Target ParamRef	
	C4A4438X.INFER_PVA		C4A4438X	C4A4438XINFER_PVA.P1	C4T4443P	C4T4443P.DACA.PV	4
	C4A4438X.INFER_CLA		C4A4438X	C4A4438XINFER_CLA.IRESULT	X015RVP	X015RVP.LAB_TEST.LABVAL	
	C4A4438X.INFER_CLA		C4A4438X	C4A4438XINFER_CLA.IRESULT.TEXTREF	X015RVP	X015RVP.LAB_TEST.LABVAL	
	C4A4438X.INFER_PVA		C4A4438X	C4A4438XINFER_PVA.IRESULT	X015RVP	X015RVP.LAB_TEST.LABVAL	
	C4A4438X.INFER_PVA		C4A4438X	C4A4438XINFER_PVA.IMIN	X015RVP	X015RVP.LAB_TEST.IMIN	
	CAAAAARWINIEED DUA		CANAATOV	CANADOVINIEED DVA IMONITU	VOLEDVO	VOLEDIAD TEST IMONITU	

When you select a CM item in the list above this, all Parameter references found in its embedded Function blocks are added to this list. The *Target ParamRef* data column is editable.

## Parameter References columns: data columns



- BlockName: The embedded Function Block name that contains this Parameter reference connection
- Expr Ref: Indicates if this Parameter reference is located in an Expression type such as an AuxCalc expression statement
- CM SourceName: The name of the Control Module that is the source of the Parameter reference
- Source ParamRef: The name of the Parameter reference in the format ControlModule.BlockName.Parameter
- CM TargetName: The name of the Control Module that is subscribing or referencing the source Parameter
- Target ParamRef: Specify a new Parameter reference connection value that is the target of the subscribed to Parameter. You do not have to change the *CM TargetName* column value.

## Parameter Value Search: edit

desc

Specify the parameter value to search for in the selected Column name from the drop-down list. In the screen capture the column name *ParamName* is selected, which filters the Parameters view to only those Parameter names that contain a value of "desc". Wildcard characters \* and ! are supported along with grouping [] brackets. See the <u>topic</u> related to filter and search for ListView type elements.

#### **Block Parameters: list**

	_				
⊿	BlockName	ParamName	ExistingParamValue	NewParamValue	
	C4A4438X	C4A4438X.DESC	PLATFORMATE PROD RVP	PLATFORMATE PROD RVP	
	C4A4438X	C4A4438X.EUDESC	psi	psi	
	C4A4438XINFER_PVA	C4A4438X.INFER_PVA.DESC	PLATFORMATE PROD RVP	PLATFORMATE PROD RVP	
	C4A4438XINFER_PVA	C4A4438X.INFER_PVA.EUDESC	psi	psi	
	C4A4438X.INIT	C4A4438X.INIT.DESC	PLATFORMATE PROD RVP	PLATFORMATE PROD RVP	
	C4A4438X.INIT	C4A4438X.INIT.EUDESC	psi	psi	
	C4A4438X.INIT	C4A4438X.INIT.EXPRDESC[1]			
	CANAGENINIT	CANAGED INIT EVENDESCIOL			

When you select a CM item in the Control Modules list, all Parameters found in the embedded Function blocks are added to this list. The *NewParamValue* data column is editable.

#### Block Parameter columns: list

_							
	BlockName	ParamName	ExistingParamValue	NewParamValue			

- BlockName: The name of the embedded Control Module Block
- ParamName: The name of the Parameter on the Block
- ExistingParamValue: The current Parameter value
- NewParamValue: Specify the new Parameter value to use for the duplicated CM

#### Search Options: toggle button



Click this header button to toggle the expanded/collapsed state for the search options. After choosing the required options, click the toggle button to collapse the options to allow specifying the filter text.

### 🥋 Clear Filter: button



Click this button to clear the current filter results in the attached ListView.

### 🔁 Data Column Names: drop-down list



This drop-down list will contain a list of column names available for filtering on the data in its attached ListView element.



## Overwrite existing files: check

Overwrite existing files

Choose this option to allow overwriting existing Control Module items in the Data View. Use this option carefully as overwritten Control Module items cannot be recovered after execution of this duplication function. To recover an accidental overwrite you must obtain the original source files and copy them back into the Project folder where these XML files reside.

13

## **Execute Duplication: button**

Execute Duplication

Click this button to begin duplicating all Control Module items that have their *Enabled* option checked.

## **MongoDB Document Server**

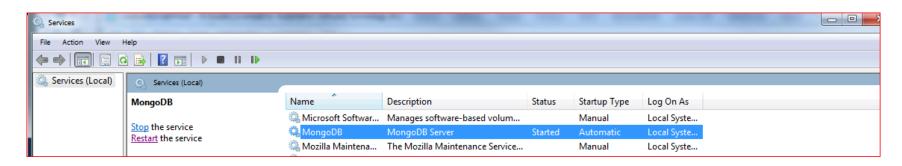
A server based document model is used by Control Architect for storage of project configuration documents such as PKS XML Configuration and HMIWeb Graphic files. This is a non-relational database typically referred to as a Document database. The database technology that was chosen is a commercial scalable product named MongoDB. During installation a local instance of this Server technology is installed. Users can choose to scale this to a Corporate Server or Cloud installation for an Enterprise model. This product does not require any additional licensing unless you choose to scale to an Enterprise model hosted by Atlas.

The Mongo Server support authentication and it is strongly recommended that users create an admin account. Administration accounts can be added to the Mongo Server database through the application <u>File-Options</u> menu. Authentication accounts exist within the MongoDB physical database that resides in your file system. If another physical database is created, you are required to create a new authentication account for the physical MongoDB Database.

**Mongo Server Options** 

## **MongoDB Server Service**

The MongoDB Server is is started/stopped through the Windows Service Control Panel. This service by default is configured to auto-start when the Computer hosting this Service is powered on and Windows becomes active.



If a problem is encountered with the Server this service may fault and require a restart. You can do this through the Windows Services Control Panel or you can choose to restart the service from the Control Architect application as shown below.



## MongoDB Server status



Indicates the status of the MongoDB Server.

This image will be displayed as red if the Server has failed.



You can right mouse-click to display a Context menu and restart the Server service.



The Service status LED image will display a blinking red image while the service is stopped and being restarted. When restarting the service, a request to stop the service is first commanded, then followed by a 10 second delay before commanding the service to start again.

## **Current Private Memory Usage**

249 MB

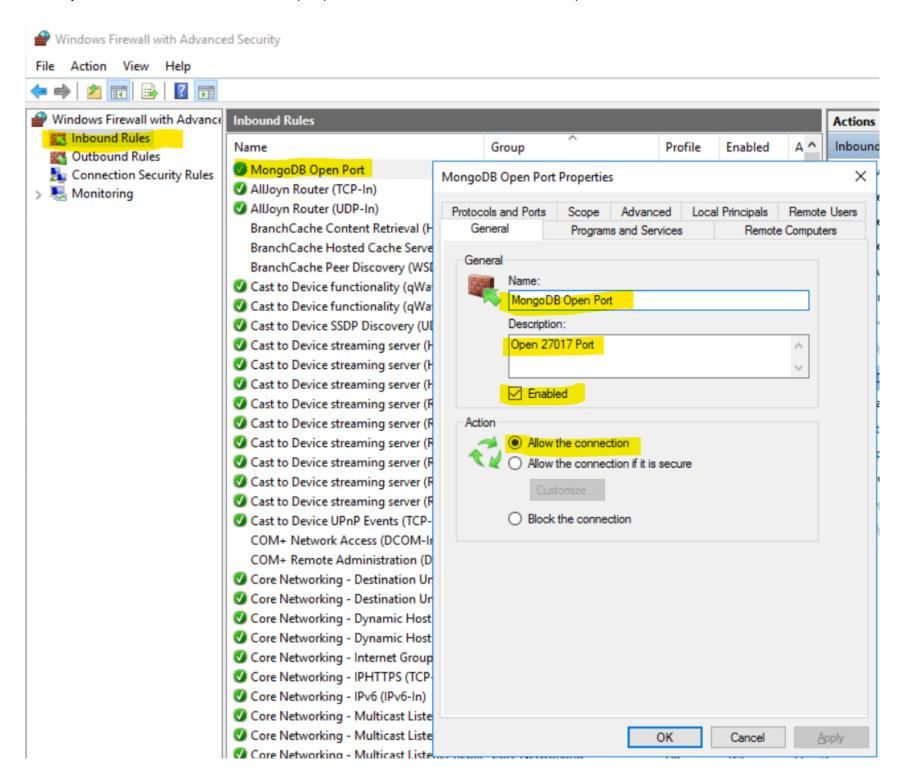
Indicates the number of Megabytes currently being used as the applications private working memory set. This usage data is updated every 5 seconds.

## **Connecting to Remote Server**

Control Architect supports connecting to a remote Server such as a Mongo Atlas Cloud host or a remote Network Server that has MongoDb server installed and running as a Service.

#### **Connecting to a remote Network Server**

The Windows Server machine hosting the MongoDB Service needs to open its port 27017 to allow inbound connections. Create a new Inbound Rule in your Windows Server 2016 server to open port 27017 to allow inbound connection requests.



#### Configure your Mongo Service configuration file similar to below:

The storage and net sections of this configuration file require adjusting to reflect your server configuration.

The default port# 27017 can be maintained or the IT specialist may choose another port to provide inbound connections.

The physical database path as shown below for "dbPath" should be changed to reflect the folder path that contains your Mongo DB database files.

```
mongod.cfg - Notepad
                                                                                           X
                                                                                     <u>File Edit Format View Help</u>
# mongod.conf# for documentation of all options, see:
    http://docs.mongodb.org/manual/reference/configuration-options/
# Where and how to store data.
storage:
  dbPath: C:\Program Files\MongoDB\Server\4.0\data
   journal:
      enabled: true
  engine:
# mmapv1:
# wiredTiger:
# where to write logging data.
systemLog:
   destination: file
   logAppend: true
  path: C:\Program Files\MongoDB\Server\4.0\log\mongod.log
# 127.0.0.1 network interfaces
net:
  port: 27017
   bindIp: 192.168.220.3 # The current IPV4 address of this VM using netstat -a
#processManagement:
#security:
#operationProfiling:
#replication:
#sharding:
## Enterprise-Only Options:
#auditLog:
```

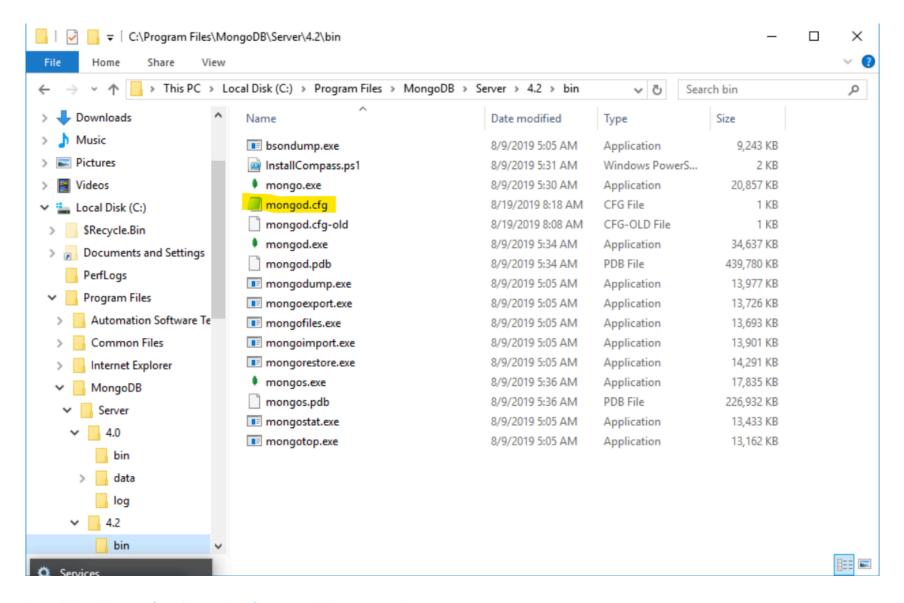
If authentication is required on the Mongo database, then modify the configuration file as follows:



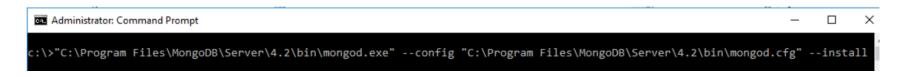
The '#' pound character is used to comment out a line in the configuration file. Also note that strict indentation is required according to yaml rules.

```
×
  mongod.cfg - Notepad
   Edit Format View
# mongod.conf# for documentation of all options, see:
    http://docs.mongodb.org/manual/reference/configuration-options/
# Where and how to store data.
storage:
   dbPath: C:\Program Files\MongoDB\Server\4.0\data
   journal:
      enabled: true
   engine:
#
  mmapv1:
  wiredTiger:
# where to write logging data.
systemLog:
   destination: file
   logAppend: true
   path: C:\Program Files\MongoDB\Server\4.0\log\mongod.log
# 127.0.0.1 network interfaces
net:
   port: 27017
   bindIp: 192.168.220.3 # The current IPV4 address of this VM using netstat -a
#processManagement:
security:
  # Enable Authentication
  authorization: enabled
  #authorization: disabled
#operationProfiling:
#replication:
```

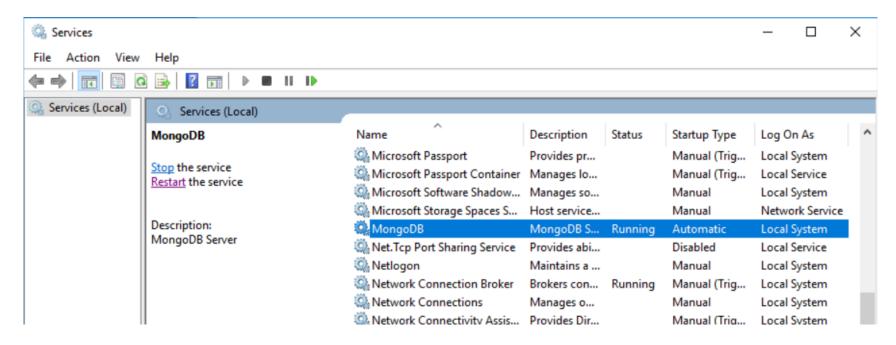
This configuration file is typically stored where the Mongo Server binary files are located, but can be stored anywhere. The Mongo Service when installed requires you to specify the location of this configuration file.



- 1. Run Command Prompt as Administrator
- 2. Enter the command script to install the Mongo Service as shown in this snapshot. Specify the folder path where the mongod.exe executable resides and also specify the folder path where your mongo configuration file resides:



After running the command to install the Mongo Service, you can view the installed Server status using Windows "service.msc" as shown below:



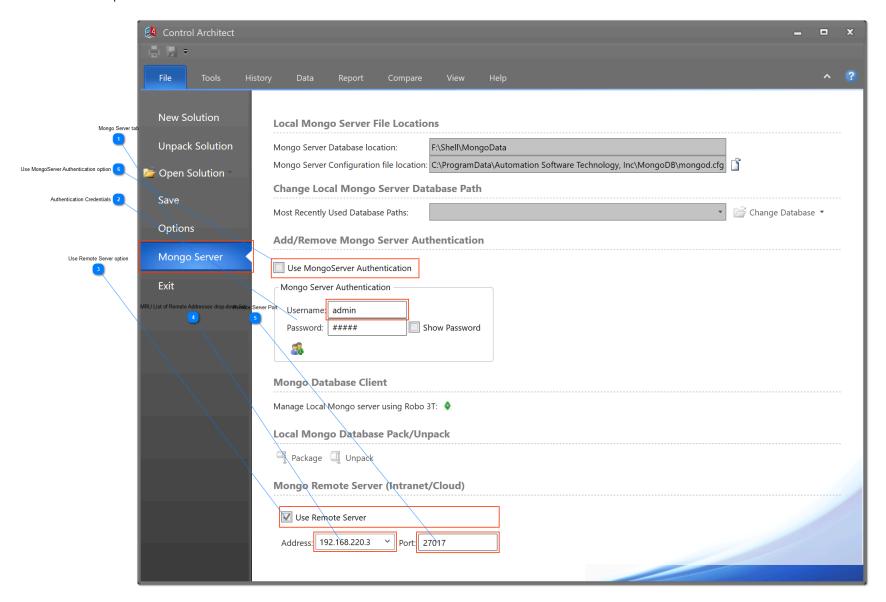
#### **Enable Server Authentication**

Determine if your Server installation will require authenticated access to the Mongo databases. If authentication is required, then your Server manager must add the users and their credentials and provide this data to the clients to allow them to connect using authentication. Refer to this topic for information about adding new users for authenticated access. Choose the <u>Use MongoServer Authentication</u> option to enable passing of authentication User Name and Password when connecting to a Mongo Server.

Administrators will want to add an admin account to the Mongo [admin] database as a "root" role. See <a href="https://docs.mongodb.com/manual/reference/built-in-roles/">https://docs.mongodb.com/manual/reference/built-in-roles/</a> for a discussion on built-in roles available for the Mongo database.

### Install a physical Mongo database

See the topic Moving Server Database for information on copying an existing physical database to move it to another file location.



Mongo Server tab

# Mongo Server

Authentication Credentials



Specify the User Name and Password for establishing an authenticated connection to the remote Mongo Server.

Use Remote Server option



Choose this option to specify a remote server. This could be a connection string for connecting to a Mongo Atlas Cloud host or an IP Address for connecting to a remote Network Server installation.

MRU List of Remote Addresses drop-down list

192.168.220.3

Enter the address of a remote Server or click this drop down button to select from a list of most recently used remote server addresses.

Remote Server Port

27017

Specify the Server port to use when connecting to a remote Server.

Use MongoServer Authentication option

Control Architect Help Documentation

Use MongoServer Authentication

Choose this option to connect to a Mongo Database using authentication.

#### **Server Maintenance Overview**

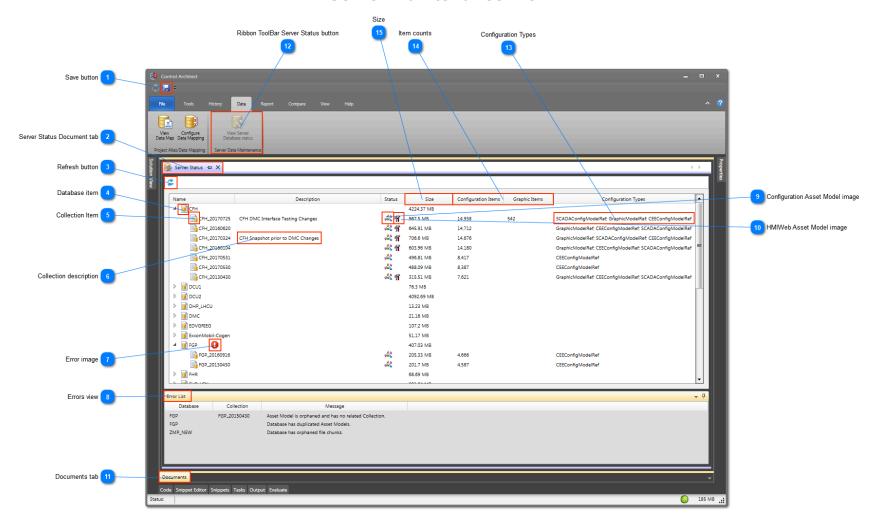
Control Architect installs with a MongoDB Server Client administration tool Compass that allows users that are proficient in MongoDB to administer your MongoDB server database. Most users will not have experience working with the MongoDB Document Server technology or understand the storage technology to execute proper maintenance activities such as deleting of a Project Database/Collection.

Control Architect provides a Maintenance View to allow users to quickly view and analyze the data consistency of Control Architect project data that is stored in the MongoDB Server database. It is recommended that all users use this view to facilitate database management functions such as deleting Databases and/or Collections.

This Maintenance View provides the following functions:

- a. View all Databases and Collections and their status that have been created as a result of importing configuration data through Control Architect
- b. Edit description values associated with Project Collections
- c. Analyze Database/Collections for data inconsistencies and repair/update when required
- d. Repair data inconsistencies
- e. Delete Database/Collection

#### **Server Maintenance View**



Save button



Click this button to save changes made to Collection description values.

Server Status Document tab



The Document tab that is created when the ToolBar View Server Database Status button is clicked.

Refresh button



Click this button to refresh the Status view. The Document Server will be queried again for all Server Database and Collections and validated for data consistency.

Database item



This item in the Tree View represents a Server Database item. Typically this would be the name of a PKS Server or Process Unit that contains multiple Collections (snapshots) of configuration data.

Collection Item



This item in the Tree View represents a Collection (snapshot) that is contained within the items parent Database. Collections represent snapshots or version of configuration data that has been imported through Control Architect.

Collection description

CFH Snapshot prior to DMC Changes

Allows you to edit or specify a description for the Collection. This description value is used throughout the application in UI to help identify the Database collection.

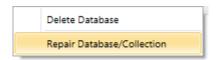
When an item is selected in the TreeView, left mouse-click to enable the edit session or press the F2 key. Press the Enter key to accept the changes or ESC to cancel.

Error image



This error image will be displayed on Database and/or Collection items if they have been found to contain a data inconsistency. Copyright © 2024 by Automation Software Technology, Inc. All Rights Reserved.

Right mouse-click to display a Context menu for the TreeView item and choose the Repair menu item to fix data inconsistencies.







Displays a list of errors that were encountered when validating the Databases and their Collections.

Configuration Asset Model image



This image status indicates that the Collection contains a Configuration Asset model.

HMIWeb Asset Model image

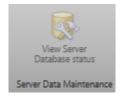


This image status indicates that the Collection contains an HMIWeb Asset model.

Documents tab

The Server Maintenance view is contained in the application Documents tab.

Ribbon ToolBar Server Status button



Click this ToolBar button to display the Server Status view.

Configuration Types

 ${\sf SCADAC} on fig Model Ref; Graphic Model Ref; CEEC on fig Model Ref$ 

Indicates the type of configuration that has been imported into this Collection. This example indicates that PKS SCADA and HMIWeb graphics have been imported into this Collection. The CEE model reference indicates that PKS compliant configuration data exists. Please note that when importing SCADA or LCN configuration data, the data is converted to PKS CEE compliant configuration and will always indicate this as a CEE model reference.

Item counts

Configuration Items Graphic Items

Displays the number of configuration or graphic items that are contained in the Collection.

Size Size

Displays the size of the Database and Collections in megabytes. The Database size is the sum of its contained Collections.

## **Managing Server Database**

Control Architect uses a document-oriented database Technology MongoDB to store configuration assets typically in the form of XML configuration files. During installation of Control Architect the MongoDB server technology was installed and deployed along with the administration tool named Mongo Compass. The Mongo Compass application will provide the tools to properly manage the document Server database.

You can access the Mongo Compass application through either Control Architect



or from your desktop Icon.

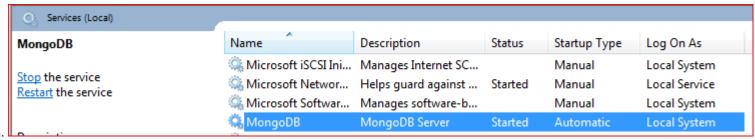
## **Moving Server Database**

You can move or copy a MongoDB Server database using the manual procedure below or by simply <u>packaging</u> the database up using Control Architect and then unpacking to a new local or remote MongoDB Server. Control Architect can both package and unpack to remotely connected Mongo servers as well as a locally hosted Mongo server.

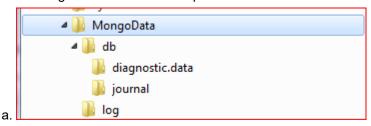
#### Manually Moving/Copying a Mongo Database

You can move the MongoDB Server database to a new location on the same machine or to another machine by following these steps below. If you are installing to a new local Host Server, you must first install Control Architect and MongoDB Server using the Control Architect installer. This will automatically install MongoDB Server if it is detected as not installed.

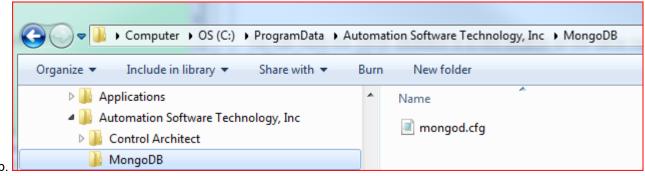
Stop the MongoDB Service



2. Locate the MongoDB Database folder path



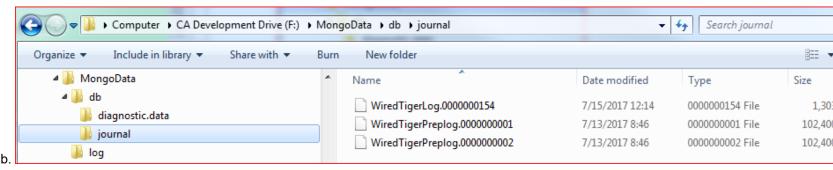
- b. Stop the MongoDB Service on the target machine
- c. Copy this root MongoData folder and its subfolders to its new location
- 3. Locate the MongoDB Configuration file
  - a. This configuration files should be located in the path "C:\ProgramData\Automation Software Technology, Inc\MongoDB"



4. Modify the existing path configuration in the mongod.cfg file to the new target paths.

```
mongod.cfg - Notepad
   Edit Format View
                                              Note how the different config options are indented a minimum
 Note: Be careful when editing this file.
                          The syntax used in this file format is known as YAML.
# of 2 space characters.
          character is used to comment out a line of text.
   Server IP-use 127.0.0.1 or 0.0.0.0 to configure for local host with default port of 27017
 bindIp: 0.0.0.0
 port: 27017
storage:
  # Dātabase File storage path
           F:\\Test\\Trond Control Architect Testing\\MongoData-10.11|.17\\db"
  dbPath:
  journal:
    enabled: true
systemLog:
  destination: file
  # Mongo Log File path
        "F:\\Test\\Trond Control Architect Testing\\MongoData-10.11.17\\log\\mongo.log"
  quiet: true
  logAppend: true
securit
  # Enable Authentication
 authorization: enabled 
#authorization: disabled
```

- b. Modify the *dbPath* and *path* variables to reflect the new target path locations. Do not change any other variable data. Maintain the 2 character tab spacing.
- 5. Delete the MongoData\Journal files
  - a. The journal files located in the *MongoData\Journal* path must be deleted. This will assure that older transactional journals are cleared when the Service is restarted.



6. Restart the MongoDB Service on both the source and target machines

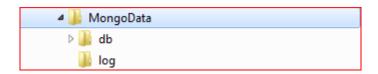
## **Maintaining Multiple Databases**

You can choose to maintain multiple MongoDB databases on your local or Network file system. A configuration file exists that provides the file path to the MongoDB database files that the MongoDB Server will use when started. By changing the paths located in this configuration file you can restart the MongoDB Server and use another physical database.

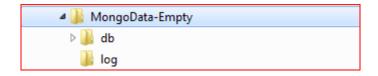


Control Architect maintains a list of Most Recently Used MongoDB Database paths that can be selected for changing the Database that is attached to the Mongo Server. You can also browse for a folder path that contains a physical MongoDB Database and create a new MongoDB physical Database path in the Mongo Server Options.

Users may choose to maintain separate databases for customer isolation. To start a new Database file you should create a new set of folder paths in your file system and change the configuration file to reflect these new paths. The folder system for a new Database location should look like the following:



In the above folder path example it is not important to name the root folder path using "MongoData", but the subfolder paths "db" and "log" should be named as in this example.



When the MongoDB Server is started using the new File path locations, it will create and maintain the database files within these folder paths.

```
mongod.cfg - Notepad
File Edit Format View Help
  Note: Be careful when editing this file. Note how the different config of 2 space characters. The syntax used in this file format is known as The '#' character is used to comment out a line of toxt
                                                           Note how the different config options are indented a minimum
 The
net:
   # Server IP-use 127.0.0.1 or 0.0.0.0 to configure for local host with default port of 27017
  bindIp: 0.0.0.0
  port: 27017
storage:
   # Dātabase File storage path
              F:\\Test\\Trond Control Architect Testing\\MongoData-10.11|.17\\db"
  dbPath:
  journal:
     enabled: true
systemLog:
   destination: file
  # Mongo Log File path path: "F:\\Test\\Trond Control Architect Testing\\MongoData-10.11.17\\log\\mongo.log" quiet: true
   TogAppend: true
security
   # Enable Authentication
  authorization: enabled 
#authorization: disabled
```

The file path locations shown in the configuration file above must be changed to point to valid file locations that exist in your file system.

- 1. Under "storage:", change the "dbPath:" file path. The file path must end with the "\\db" path component.
- 2. Under "systemLog:", change the "path:" file path. The file path must end with the "\\log\\mongo.log" path and file component.



When specifying path delimiter characters you must specify the double '\\' path delimiter character set.



The '#' character serves as the beginning of a comment that will not be read by the Mongo Server.

It is strongly recommended that your MongoDB databases be secured using authentication. Under the "security" keyword the statement: "authorization: enabled" enables authentication in the Mongo Server when restarted. If authentication is enabled, then the users of any MongoDB physical databases must create a new administration account using Control Architect Mongo Server-Options. To disable authentication, you must insert the '#' comment character in front of the "authorization: enabled" statement and remove the '#' character prefixing the "authorization: disabled" statement.

When making changes to file locations in this file you must restart the MongoDB Server. You can restart this from within Control Architect using the MongoDB Status image in the Status Bar or alternatively use the Services.msc management view in Windows to locate the Service and restart it.

## **Working with Mongo Shell**

MongoDB provides a shell interface through the Command Processor window that allows administrators and users to execute functions through the Mongo Server for maintaining your physical databases.

### **Create a User Account**

This procedure guides you in creating User type accounts on an existing authenticated Mongo Server.

Open a Command Process window and log into the Mongo Server using authentication:

```
Collaboration Windows Command Processor - To Program Files Mongo DB Server bin Nongo exe" localhost -u admin -p admin --authentication Database admin --authentication Mechanism SCRAM-SHA-1

D:\"D:\Program Files Mongo DB Server bin Nongo exe" localhost -u admin -p admin --authentication Database admin --authentication Mechanism SCRAM-SHA-1

Mongo DB sell version v3.4.6
connecting to: nongo db:/127.8.0.1:27017/localhost
Mongo DB server version: 3.4.6
vuce admin --authentication Database admin --authentication Mechanism SCRAM-SHA-1

**Connecting to: nongo db:/127.8.0.1:27017/localhost
Mongo DB server version: 3.4.6
vuce admin --authentication Database admin --authentication Mechanism SCRAM-SHA-1

**Connecting to: nongo db:/127.8.0.1:27017/localhost
Mongo DB server version: 3.4.6
vuce admin --authentication Database admin --authentication Database admin --authentication Database admin --authentication Mechanism SCRAM-SHA-1

**Connecting to: nongo db:/127.8.0.1:27017/localhost
Mongo DB server version: 3.4.6
vuce admin --authentication Database admin --authentication Database
```

The login command: "D:\Program Files\MongoDB\Server\bin\mongo.exe" localhost -u admin -p admin --authenticationDatabase admin --authenticationMechanism SCRAM-SHA-1 logs into the Mongo Server that is currently configured for authentication. Replace the *admin* password and *admin* user name with the credentials you created when you enabled authentication. Do not change the authentication mechanism that is used as Control Architect will only authenticate using this type. If your Mongo Server was installed to the "C:\" folder path, then replace the 'D' path with your 'C' folder path. Do not change the authentication database "admin". Mongo will require authenticating against the [admin] database.

Enter the command: "use admin" as shown above to target the [admin] database, which is where all authentication will authenticate against.

Enter the command: "show users" as shown above to list the current user accounts configured for this MongoDB physical database. In the example above you can see 2 existing user accounts that have been created. The "admin" user has been granted the "root" security access role. The "user" user account has been granted the "readAnyDatabase" security access role.

#### Create a new user account type

Enter the command: db.createUser({user: "user1", pwd: "user1", roles: [ "readAnyDatabase" ] })

This will create a new user type account with the User name="user1" and the Password="user1".

Enter the command: "show users" to verify the new user account that was created as shown above.

Copyright © 2024 by Automation Software Technology, Inc. All Rights Reserved.

To verify the new user account credentials and access, type "exit" to exit the Mongo Shell and log back in using the new user account credentials as shown below. Type the command text "show dbs" to verify that the user account has access to the required databases.

```
:\Program Files\MongoDB\Server\bin\mongo.exe" localhost -u user1 -p user1 --authenticationDatabase admin --authenticationMechanism SCRAM-SHA-1
ting to: mongodb://127.0.0.1:27017/localhost
8 server version: 3.4.6
admin
 use aumin
vitched to db admin
show dbs
PS4
R430_CHARTUIEW_UISUALS_TESTING
```

#### Drop an existing user account

```
\Program Files\MongoDB\Server\bin\mongo.exe" localhost -u admin -p admin --authenticationDatabase admin --authenticationMechanism SCRAM-SHA-1
shell version v3.4.6
ing to: mongodb://127.0.0.1:27017/localhost
server version: 3.4.6
use admin
witched to db admin
db.dropUser("user1")
           "_id" : "admin.admin",
"user" : "admin",
"db" : "admin",
              id": "admin.user",
                                     "role" : "readAnyDatabase",
"db" : "admin"
```

Enter the command: db.dropUser("user1") to delete the previously created user account. Please note that all Mongo Shell commands are case

Enter the command: "show users" to verify the deletion of the user account as shown above.

Enter the command: "exit" to exit the Mongo Shell and disconnect from the Mongo Server.

Use this <u>reference</u> for more information on working with the Mongo Shell.

#### Repair a corrupted Mongo Database

This procedure guides you in repairing a corrupted Mongo Database that exists in a Mongo Server. Corruption is often the result of an unexpected shutdown or termination of the Mongo Server while database activity was present or during Database migration activities. It is strongly recommended to take periodic backups of the Mongo database using the Package function.

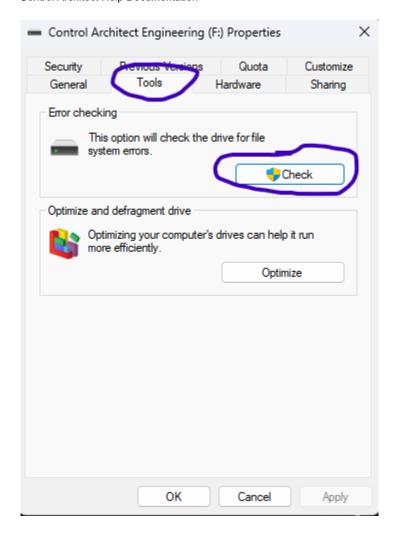
#### **Delete existing Mongo Lock file**

If the Mongo Database was not shutdown properly there will be a file named "mongod.lock" in the Mongo physical database folder. If present this file must first be deleted. It would be located for example in the database path "F:\MongoData\db".

#### Verify your windows file system is not corrupted

Run Check Disk on the system drive where your MongoDB physical database resides.

Right mouse-click on the system drive to view the disk properties. Select the "Tools" tab and click the "Check" button target to begin scanning your system drive for errors.



#### **Open a Command Process window and Run As Administrator**

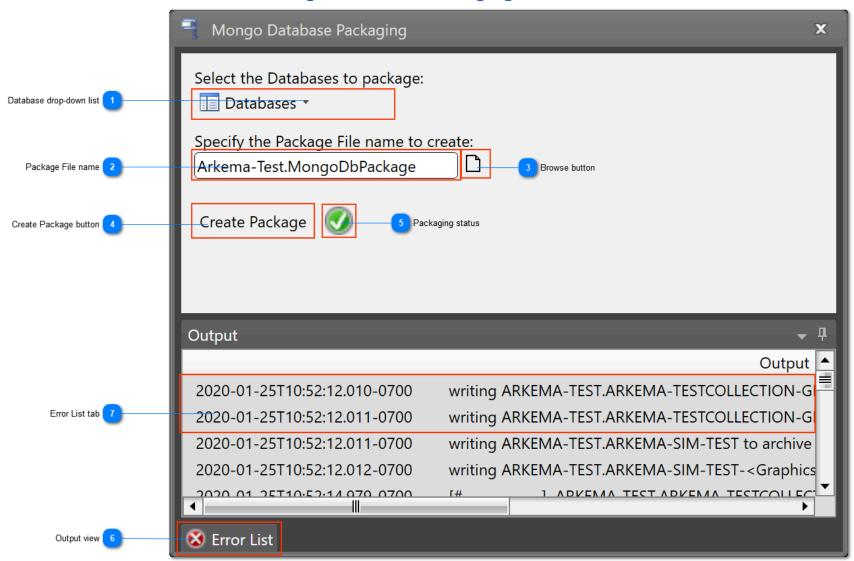


The Mongo Shell is executed using "C:\Program Files\MongoDB\Server\bin\mongod.exe". You must enclose the command path in double quotes to encapsulate space characters in the file path. Adjust the path if the Mongo binaries are located in another folder path.

The --dbpath shell command indicates that the Mongo Database is found in the following file path location "F:\MongoData\db". Again ensure path is enclosed within double quote chars.

The --repair shell command will attempt to repair the physical Mongo Database and all its contained collections, which includes rebuilding all index files.

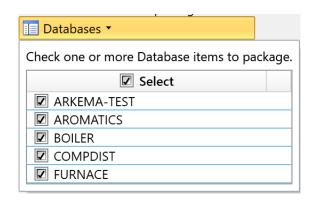
## **Mongo Database Packaging window**



## Database drop-down list



The current list of logical databases that are found in the connected Mongo Server database. Select one or more databases to package for deployment.



Package File name

# Arkema-Test.MongoDbPackage

Click the Browse Button to specify the Package File location and name.

## Browse button



Click this button to specify the Package File location and name.

## Create Package button

# Create Package

Click this button to begin creating the package file.



## Packaging status



Shows the status of the Package operation.



## **Output view**

2020-01-25T10:52:12.010-0700

2020-01-25T10:52:12.011-0700

writing ARKEMA-TEST.ARKEMA-TESTCOLLECTION writing ARKEMA-TEST.ARKEMA-TESTCOLLECTION

Click the Output tab to view messages associated with the packaging operation. These messages are returned by the MongoDB Server during a Mongo Dump command from the connected Server.

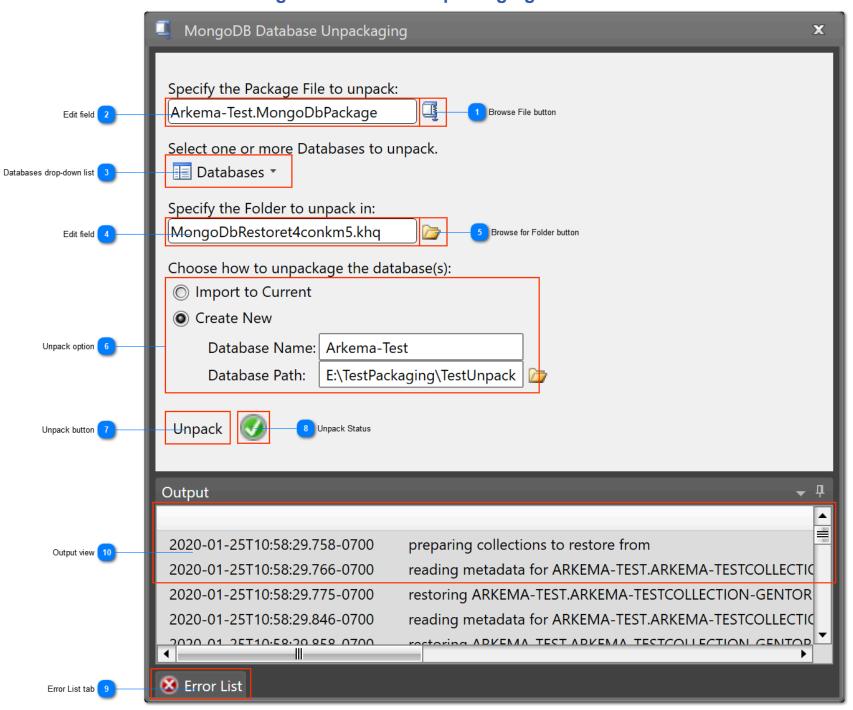


#### **Error List tab**



Click this Tab or hover with your mouse cursor to view any errors associated with the packaging operation. If any errors were encountered, this tab will be flashing red.

## **MongoDB Database Unpackaging window**



Browse File button



Click this button to select a Package file to unpack.

Edit field

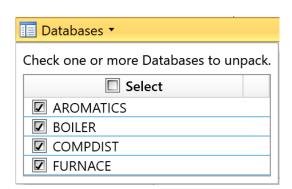
# Arkema-Test.MongoDbPackage

The name of the Package file to unpack. Click the Browse button to select a Package file.

🔧 Databases drop-down list



Select the databases to unpack. This drop-down list contains a list of databases that are contained in the selected Package file.





#### **Edit field**

# MongoDbRestoret4conkm5.khq

The default temporary folder path where the package contents will be extracted to. Click the Browse button to navigate to a folder path where the Package file contents will be extracted to during the unpack operation. This folder path and its contents are deleted after the unpack operation has completed.

5

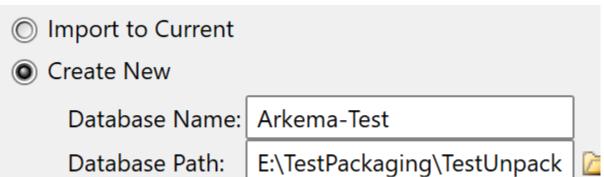
## Browse for Folder button



Click this button to browse for a folder to extract the package contents to. This path is defaulted to your ...Local\Temp folder for your account profile.

6

## **Unpack option**



Choose how to restore the selected Mongo databases. You can either restore/import to the current connected Mongo Database or create a new physical Database path to contain the new logical databases.



You cannot create a new physical Mongo database path for a remotely connected Mongo Server. This option only applies when the Mongo Server is locally hosted on your PC.

7

#### **Unpack button**

## Unpack

Click this button to begin the Unpack operation. You can click this button again during the unpack operation to cancel it.

8

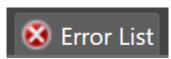
#### **Unpack Status**



Indicates the status of the Unpack operation.

9

#### **Error List tab**



Click this Tab or hover with your mouse cursor to view any errors associated with the unpack operation. If any errors were encountered, this tab will be flashing red.

10

#### **Output view**

2020-01-25T10:58:29.758-0700 preparing collections to restore from reading metadata for ARKEMA-TEST.ARKEMA-TESTCOLLECTIC

Click the Output tab to view messages associated with the unpack operation. These messages are returned by the MongoDB Server during toommand from the connected Server.

## **Project Data Map**

Control Architect provides a feature that allows mapping of multiple version sets of PKS Xml folder files to a common project alias name. It accomplishes this mapping of multiple version files by parsing the last folder name in a folder path to form a common project alias name.

This view can be accessed from the Ribbon Toolbar menu Data/View Data Map.

#### **Project Alias Name Formation**

Typically you will define an internal standard as to how Xml compressed file names are formed when exporting the configuration from a server. An example would be where a Coker unit export would have its folder name (compressed file name) formed as in "DCU\_20130628", which indicates that this is a Coker type unit exported on June 28, 2013. Another version from the same unit server would be formed as "DCU\_20130726". The common alias name that can be formed using this convention would be "DCU".

Control Architect uses Regex expression types to allow a user to provide a custom match expression that will accurately resolve a file/folder name to a common project alias name. The default expression installed with Control Architect simply looks for the first '\_' character and uses all characters to the left of '\_' to form the common project alias name.

#### **Default Regex Match Expression**

This is the default expression that will search for the first '\_' character and use all characters to its left to form the project alias name:

```
(^[a-zA-z0-9]+)[]
```

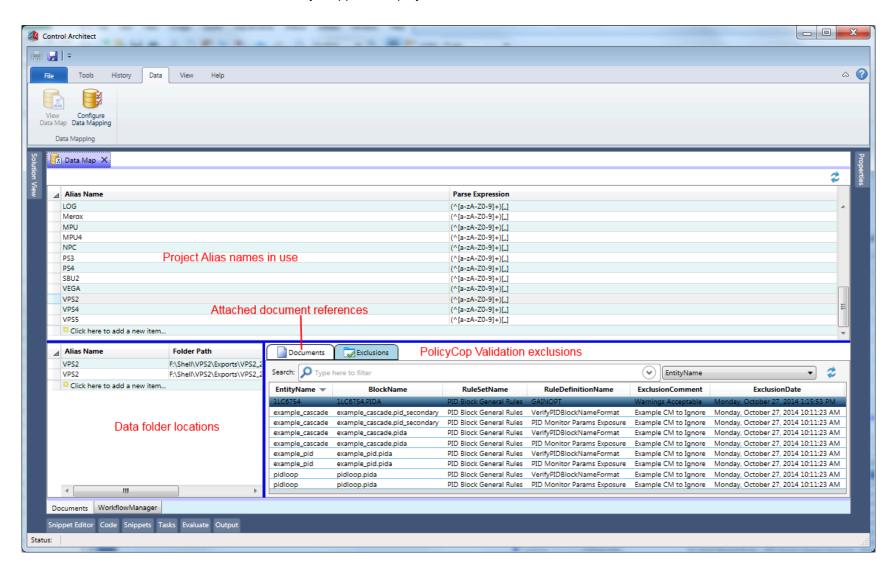
This expression simply tries to match starting at the beginning of a text any Upper/Lower case letter including the digits 0-9 until the '\_' character is encountered. The resulting match if this expression was applied to the text phrase "DCU2\_20130628" would be "DCU2".

It is recommended that users form the folder name that contains PKS Xml configuration files using the following format: **UUUU\_yyyymmdd** where U=Unit descriptor such as DCU, y=calendar year, m=calendar month and d=calendar day.

## **Using the Project Data Map**

The Project Data Map view shows 3 groups of information that are currently used in a project data map.

- 1. The Project Alias names in use
- 2. The Xml folder locations that are mapped to a project alias name
- 3. The document references that are currently mapped to a project alias name



#### **Project Alias names**

This data grid displays the current set of Xml project alias names that have been formed as a result of attaching a document reference using the ChartView function on a function block. The following functions are available within this data grid:

#### **Add New Xml Alias Name:**

Click the data grid row that indicates "Click here to add a new item..." to manually add a new Xml alias name. You do not normally have to manually add project alias names here as this will automatically be defined and added when you view a particular Control Module in the ChartView and add a document reference to a function block.

#### **Edit Existing Xml Alias Name and Regex Parse Expression:**

Clicking in a data grid cell that for a selected row will enable editing of the cells contents. Click the toolbar **Save** button to save any changes.

If the Regex parse expression is changed, it is recommended that you test the expression modification by using the <u>Project Data Map</u> <u>Configuration</u> to verify the results of the modifications. Simply copy the modified expression, navigate to the Data Map Configuration, paste the copied expression into the current default Regex expression text box, enter a test phrase and click the Test button. Do not click the Save button unless you want to keep this Regex expression as the application default to use when Xml alias names are automatically defined as a result of adding an attached document to a new PKS Xml configuration file folder.

#### **Delete Xml Alias:**

Press the **Delete** key to delete the selected data grid row item. After confirming the delete request, if document references exist for the project alias you will be presented with a view to choose how to resolve the disposition of its attached documents. You can choose to delete the document references or to move the document references to another project alias mapping. Only a single Xml project alias data grid row item can be deleted in the same operation. If multiple data grid rows are selected, the first data grid row that was selected in the sequence will be the item that is deleted.

Tip: If the format used for defining folder path names is changed and another Xml project data map alias is automatically defined, then you can delete the old project alias and choose to move its attached document references to this new Xml project alias.

#### **View Attached Document References**

You can view attached document references for a single or multiple project alias names by pressing and holding the **Control** Key while selecting an Xml alias name row in the data grid. Pressing **Ctrl+A** while focused in the Xml project data map grid will select all configured Xml alias names and their <u>attached document references</u>.

#### **Delete Attached Document References**

You can delete attached documents by selected one or more document references and pressing the Delete key.

## Tips and Tricks

• If the format used for defining folder path names is changed and another Xml project data map alias is automatically defined, then you can delete the old project alias and choose to move its attached document references to this new Xml project alias.

#### **Shortcuts**

• Pressing the key sequence Ctrl+A while focused in a data grid or list will select all its contained items.

#### Notes:

- You cannot edit or delete any data displayed in the Xml folder locations list.
- Only a single Xml project alias data grid row item can be deleted in the same operation.
- If this view is currently open when a document reference is attached while in a ChartView, this view will automatically be refreshed. You can can also manually refresh the view by clicking the refresh button located in the upper right corner of the view.

### **Project Data Map Options**

The default Regex expression that is used when a new Project data alias is created can be changed by the user through the <u>Project Data Map Configuration</u> view.

#### Internet References to Regex Expression Development

Copyright © 2024 by Automation Software Technology, Inc. All Rights Reserved.

- Regular Expression Examples
- RegexOne Learn Regular Expressions
- RegexLib.com

## **Project Data Map Options**

This view allows you to change the default Regex expression that is used to parse a folder name where PKS configuration Xml files reside to form a common project alias name.

This view can be accessed from the Ribbon Toolbar menu Data/Configure Data Mapping.

### **Default Regex Match Expression**

This is the default expression that will search for the first '\_' character and use all characters to its left to form the project alias name:

```
(^[a-zA-Z0-9]+)[_]
```

This expression simply tries to match starting at the beginning of a text any Upper/Lower case letter including the digits 0-9 until the '\_' character is encountered. The resulting match if this expression was applied to the text phrase "DCU2\_20130628" would be "DCU2".

## **Regex Test Phrase**

Enter a typical folder name you will use to identify a version set of PKS Xml configuration files. An example would be **DCU2\_20131023**. This folder name would be part of a folder path such as "C:\Refinery\PKS\Exports\DCU2\_20131023". This is test phrase is used to evaluate the Regex expression to verify that a match can be found that will form the alias name.

#### **Test Result**

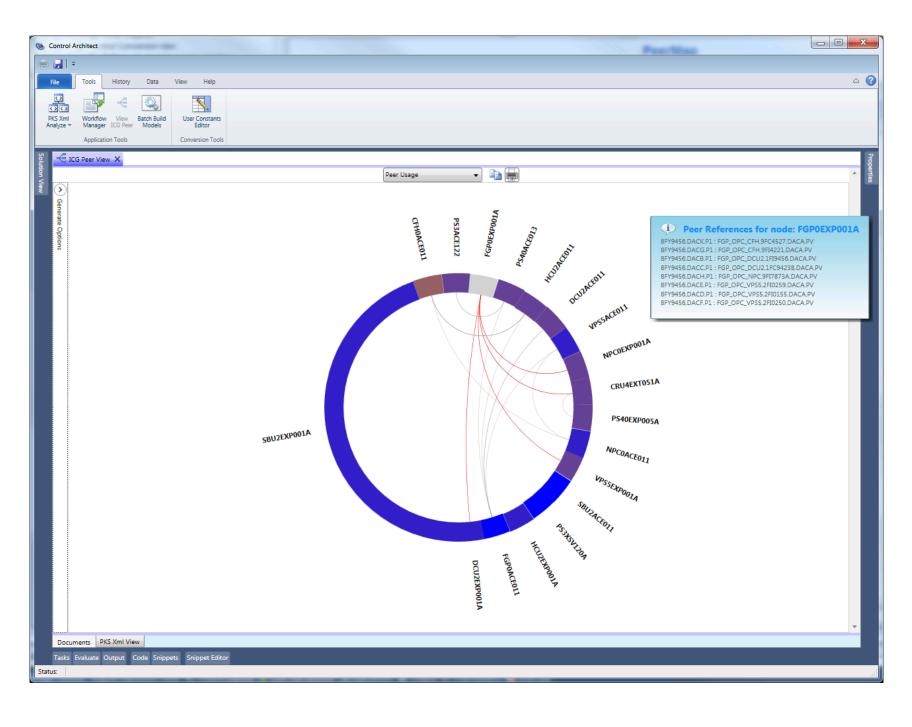
Clicking the **Test** button will evaluate the Regex expression using the Test Phrase contents as the input to form an alias name. If the evaluation succeeds the result will display "Passed: DCU2" where the text component DCU2 indicates the parsed alias name that was formed.

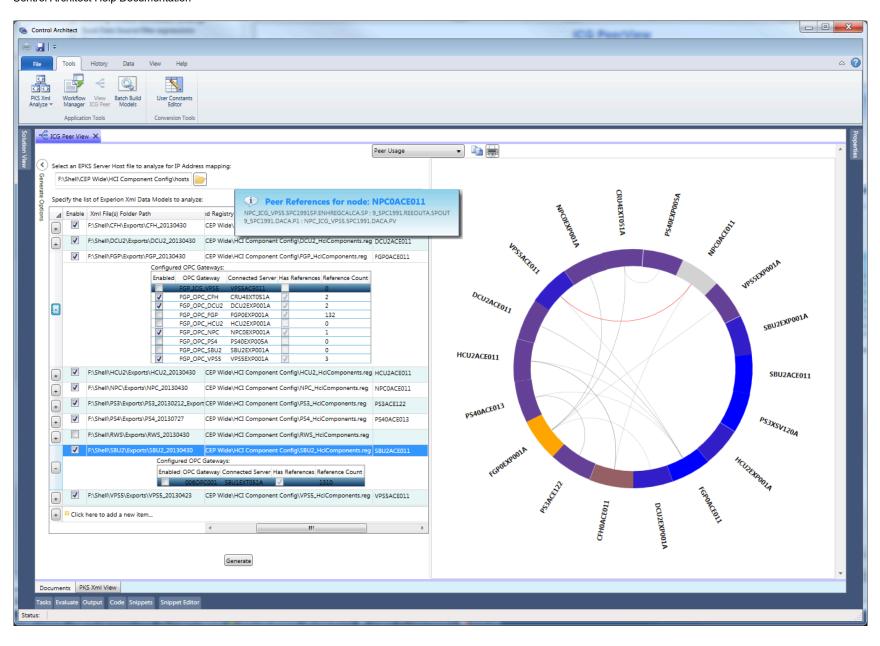
#### **Save Expression**

Click the Save button to save the modified Regex expression and exit the options view.

#### **ICG PeerView**

InterCluster Peer can be visualized using the ICG PeerView tool.





## **Configure the View**

The InterCluster Gateway (ICG) view requires information from the EPKS Server in order to analyze and determine the Gateways and their types that a Server is configured for. Data is required from the Registry and the Hosts file to successfully configure the ICG Peer analysis. The Hosts file provides the configured IP addresses that provide the mapping between Nodes and Gateways for the Server.

Configuration data that defines the Peer Map view is saved between sessions. Simply click the **Generate** button to again generate the visual ICG Peer Map view when reinvoked.

Configuring the ICG View for analysis requires the following data:

- 1. Exported HciComponents Registry from each PKS Community Server.
- 2. A single hosts file that defines the network IP communications
- 3. Exported C300/ACE configuration from each PKS Server that is imported into a Control Architect supported MongoDB database.

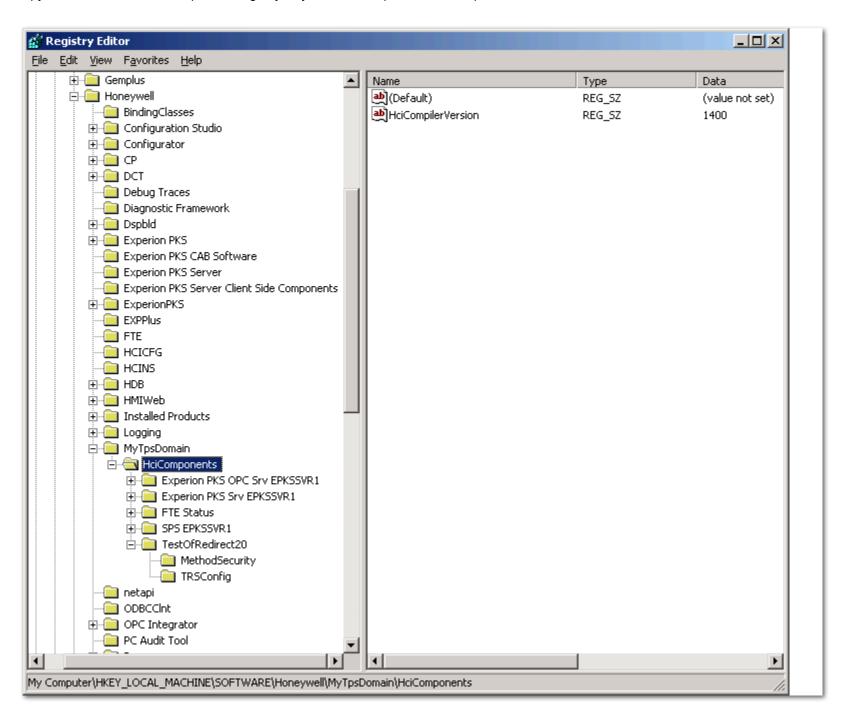
## **Export Server Data**

The InterCluster Gateway (ICG) view requires information from the EPKS Server in order to analyze and determine the Gateways and their types that a Server is configured for. Data is required from the Registry and the Hosts file to successfully configure the ICG Peer analysis. The Hosts file provides the configured IP addresses that provide the mapping between Nodes and Gateways for the Server.

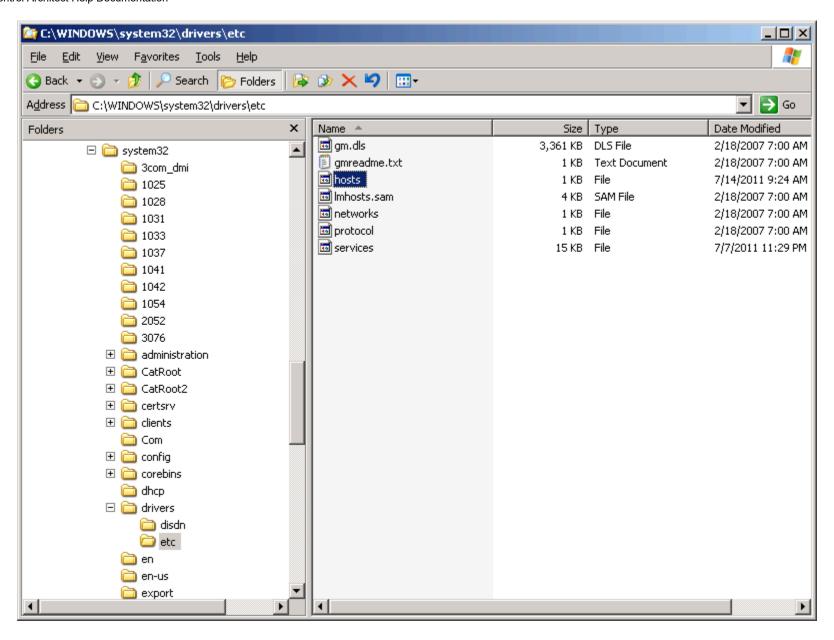
Invoke the Registry editor from your PKS A Server using the Run Command text "RegEdit.exe".

The Registry data that is required is the Registry key *HciComponents*. This can be found in the Registry hive *HKEY\_LOCAL\_MACHINE* \SOFTWARE\Honeywell\MyTpsDomain\HciComponents or *HKEY\_LOCAL\_MACHINE\SOFTWARE\WOW6432Node\Honeywell\MyTpsDomain* \HciComponents depending on the PKS Release where the registry key WOW6432Node is applicable to 64 bit PKS Servers R410+

- 1. To export this data select the Registry key *HciComponents* as show below and right mouse-click to display a shortcut menu. Select the menu item **Export** to export the configuration to an external \*.reg file. Specify the file name as unique to the Server as in *DCU1\_HciComponents* to help differentiate the owner of this Registry key data.
- 2. Locate the Hosts file, which can be found in the file system path *C:\WINDOWS\system32\drivers\etc.* Locate the file named *Hosts* with no file extension.
- 3. Copy the Hosts file and the exported Registry Key file to a Jump Drive to transport the files to reference within Control Architect.



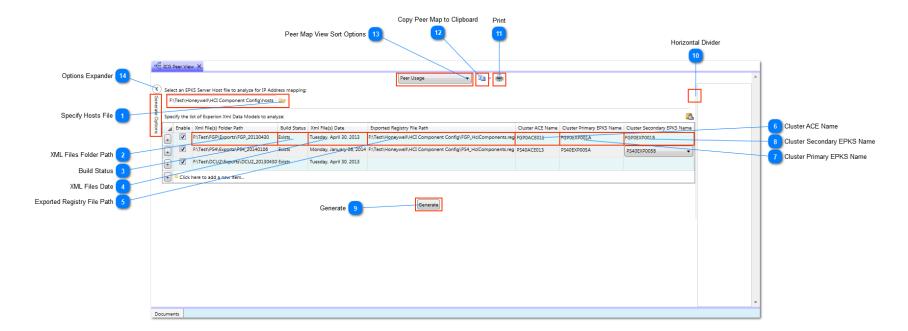
Location of the Hosts file can be in the file path below:



# Configure the peer nodes

You can configure the peer nodes by specifying each EPKS Server node and referencing its data.

To copy the data from the Batch Build view, click the button to copy the list of EPKS XML configuration project references to the Windows Clipboard. Click the **Paste** button in the peer view to paste the Windows Clipboard contents into the view. You will then have to manually reference the ACE and EPKS node names from the drop-down lists.



Specify Hosts File

F:\Test\Honeywell\HCI Component Config\hosts

Click the browse button to locate the Hosts file that was exported from the EPKS Server.

XML Files Folder Path

F:\Test\FGP\Exports\FGP\_20130430

Click the Browse button to locate the folder that contains the exported XML configuration files from an EPKS Server. Click the DataGrid row to select the row. Click again in a cell to expose the Browse button.

Build Status

Exists

The current generation status will be displayed here.

XML Files Date

Tuesday, April 30, 2013

When a reference to an XML configuration project is added, the file date of the exported EPKS XML configuration files is displayed here.

Exported Registry File Path

 $F:\label{lem:component} F:\label{lem:component} F:\label{lem:components} F:\label{lem:componen$ 

Click the Browse button to locate the exported Registry Key file for the EPKS Server that contains the XML configuration files.

Cluster ACE Name

FGP0ACE011

Click in this cell to expose a drop-down list that contains all nodes found in the attached Hosts file. Select the node item that represents the ACE node.

Cluster Primary EPKS Name

FGP0EXP001A

Click in this cell to expose a drop-down list that contains all nodes found in the attached Hosts file. Select the node item that represents the primary EPKS Server node.

Cluster Secondary EPKS Name

FGP0EXP001B

Copyright © 2024 by Automation Software Technology, Inc. All Rights Reserved.

Click in this cell to expose a drop-down list that contains all nodes found in the attached Hosts file. Select the node item that represents the secondary EPKS Server node.



Click this button to begin generating the ICG Peer Map view based on the configured EPKS Server data.

Horizontal Divider

Place your mouse cursor over this divider image and left mouse-click to drag the size of the view.

Print

Click this button to print the Peer Map view to an XPS Document file.

Copy Peer Map to Clipboard

Click this drop-down button to copy the Peer Map view and its data to the Windows Clipboard. You can also choose other options including how data is sorted when pasted into Excel and whether to copy data for only Gateways enabled for visual analysis.

Peer Map View Sort Options

Peer Usage

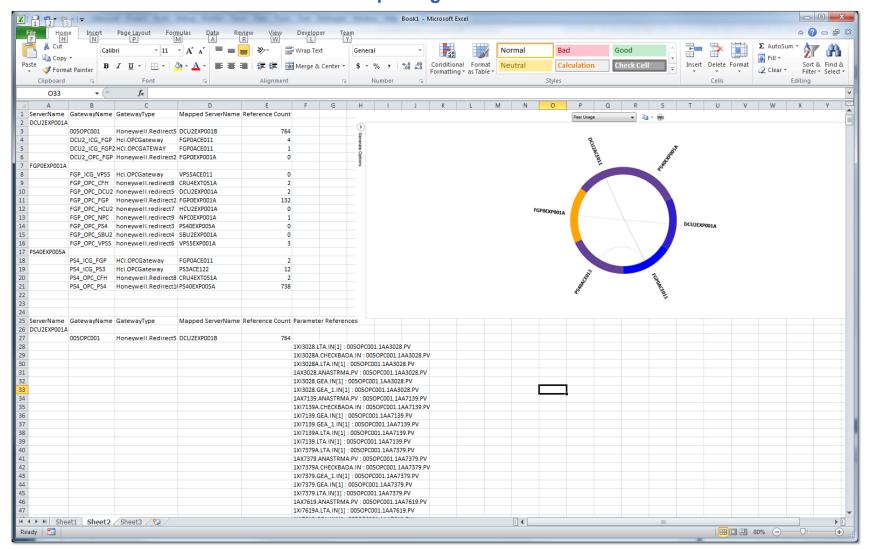
Click this drop-down list to select the type of sort to apply to the visual Peer Map view.

Options Expander

Generate Options

Click this button to toggle the visibility of the generation options.

# **Exporting to Excel**



Peer Map visual and data can be exported to Microsoft Excel. Click the **Copy** button to copy the Peer Map data to the Windows Clipboard. Use the **Paste Special** function in Excel to paste both the bitmap object and the text object into your Excel Spreadsheet.

# **Workflow Manager**

### **Overview**

The Workflow Manager manages views for various Workflow type activities including Policy Validation, DataSource to Excel Conversion, Bulk Build/Edit, and Control Conversion project files generation.

# **PolicyCop**

PolicyCop is an application tool that analyzes Process Control Strategy configuration and reports errors and other information about the configuration, such as suggested best practices, misuse of function blocks, range scaling issues, missing or incorrectly chosen options, standardization failures, etc. Many of the issues concern design violations and best practices in the design and use of functions that define a Control strategy.

PolicyCop is intended for engineers and organizations that want to develop process Control strategies where standards are enforced and to minimize process Startup issues typically found with new Control Strategy developments.

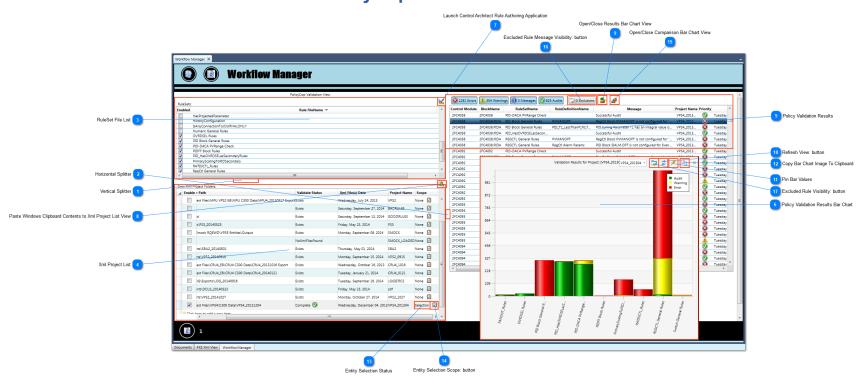
This release of Control Architect supports validation of Honeywell PKS C300 based configuration.

# **Getting Started with Policy Validation**

PolicyCop analyzes configuration elements typically found in Control Modules, known as targets, by using rules that return informational messages about the targets when the rules are violated. A report that contains the informational messages appears in the user interface. Messages identify any relevant configuration and design issues and, when it is possible, supply information about how to fix the target. A message is associated with both a specific rule and a specific target and can be excluded from further analysis. PolicyCop provides a default set of rules. You can create additional custom rules by using the <u>CARuleAuthor</u> application. A workflow session is used to specify the Xml configuration Projects, the set of entities you want to analyze, the rules that are used to analyze the entities, and the excluded messages.

- 1. Unzip the downloaded PolicyCop Rule sets to a folder on your system. In Control Architect click the **File** tab in the main Toolbar and set the path to the unzipped Rules Folder.
- 2. Click the **Tools** tab and click the **Workflow Manager** button.
- 3. Click the **Workflow** button to add a PolicyCop view to the Workflow Manager view. Your Rule sets should be visible in the RuleSets ListView. If necessary drag the vertical divider to the right of the Rule sets ListView to adjust your view contents. You can also adjust the horizontal view divider to adjust the contents size of the Rule sets ListView and the Xml Projects DataGrid.
- 4. In the DataGrid titled *Cxxx Xml Project Folders*, click the \* new row indicator to add a new project item. Click in the row cell labeled *Xml File(s) Folder Path* to expose a folder browse button and select the folder where your XML configuration files reside. It will look for the Configuration Asset Model that was generated when you added a new PKS Xml Project above.
- 5. Enter an alias name for the Project in the Project Name column cell.
- 6. Click the button in the Scope column cell to select the scope of Control Module entities to validate.
- 7. Select one or more Rule sets to use for the validation task.
- 8. Click the **Execute Validation** button to start the validation task.

# **PolicyCop Workflow View**



# Vertical Splitter

Use this splitter to resize the results view width.

Porizontal Splitter

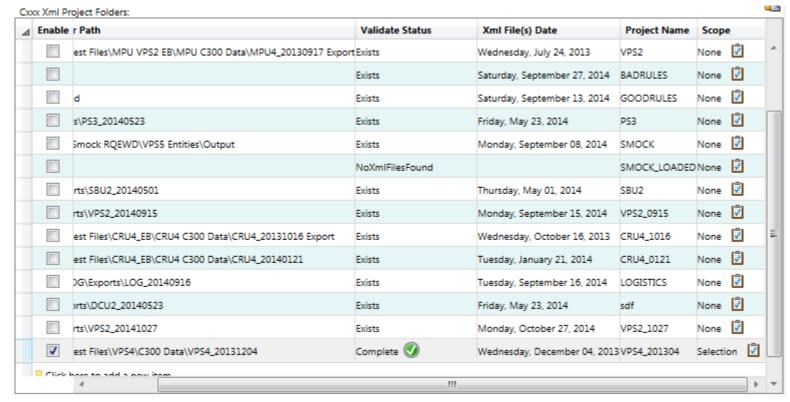
Use this splitter to resize the RuleSet and Xml Project view components height.

RuleSet File List



This is the list of available RuleSet files in the specified folder path.

Xml Project List



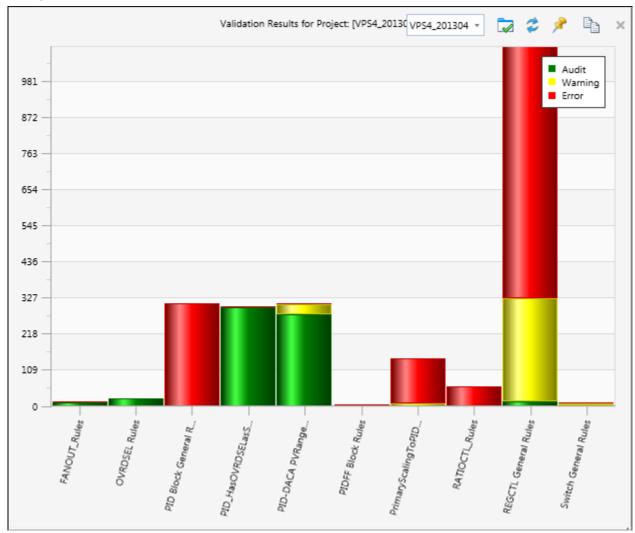
This is the list of Xml file folders that contain PKS Xml configuration files and their Configuration Asset data Model.

# Policy Validation Results



The Results view contains the messages emitted by the rules when executing policy validation. You can copy and paste these results into Excel using **Ctrl+C/Ctrl+V** keys. You can navigate to the Control Module Chart View by left mouse double-clicking on a result item.

### Policy Validation Results Bar Chart



This is a graphical bar chart that indicates the results of the last executed policy validation.

# **The Figure 2** Launch Control Architect Rule Authoring Application



Click this button to launch the Rule authoring application that is used to develop PolicyCop rules and their definitions.

# Paste Windows Clipboard Contents to Xml Project List View



Click this button to paste the Windows Clipboard contents to auto-build the list of Xml configuration projects. It is recommended to maintain your list of PKS Xml configuration file projects in the Batch Build Models view. This Batch Build Models view provides a function that allows you to copy its current Xml project list to the clipboard for pasting into other compatible Xml project list views.

# Open/Close Results Bar Chart View



Click this button to toggle the visibility of the Results Bar Chart view. If the bar chart is visible when an entity item is navigated to in a Chart View, then it will be automatically hidden from view. Click this button to toggle it visible again.

### Refresh View: button



Click this button to refresh the Bar Chart view.

### Pin Bar Values



Toggle this button to pin/unpin the Bar Chart bar values.

### 🔁 Copy Bar Chart Image To Clipboard



Click this button to copy the Bar Chart View image to the Windows Clipboard for pasting into Excel.

# Entity Selection Status



Hover your mouse cursor over this status text to view a ToolTip that indicates the count of selected entities to process. If more than 0 entities are selected, then this status will indicate "Selection".

### Entity Selection Scope: button



Click this button to display an Entity Selection view that allows you to select entities from the current Xml project for validation.

# Open/Close Comparison Bar Chart View



Click this button to toggle the visibility of the Comparison Results Bar Chart view. If the bar chart is visible when an entity item is navigated to in a Chart View, then it will be automatically hidden from view. Click this button to toggle it visible again. This Bar Chart view provides results from the last 2 configuration projects that were validated showing the increase/decrease in quality metrics.

### Excluded Rule Message Visibility: button



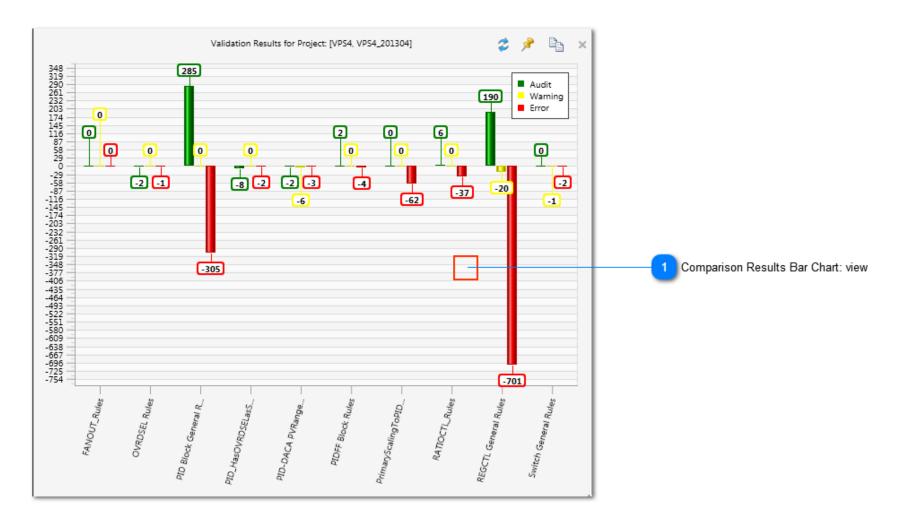
Toggle this button to show/hide Rule Message items that are currently marked as excluded. Rule Message items that are excluded from a particular Rule validation do not reduce the quality metrics, but can help the metrics by marking the entity item as a valid exception to the Rule.

#### Excluded Rule Visibility: button



Toggle this button to include/exclude Rule Message items that are currently marked as excluded when visualizing the metrics in the stacked Bar Chart view. Rule violations that are marked as excluded will improve the quality metrics when not visible.

# **Comparison Results View**



# Comparison Results Bar Chart: view

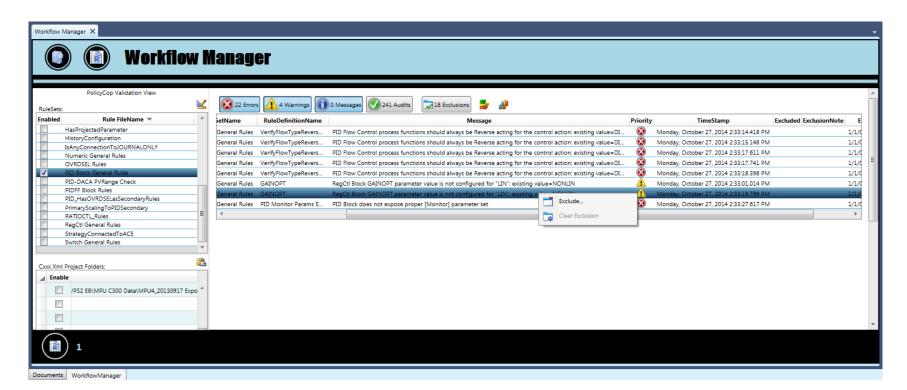
This Bar Chart view shows the results comparison of the last 2 executed PolicyCop validations. This comparison view is used to show the change in quality metrics between subsequent versions of a configuration project.

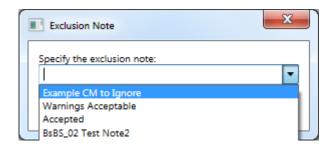
# **Using Rule Exclusion**

When validating project configuration there may be rule violations that occur for specific Control Module strategies that are technically acceptable, but for the majority of the Control Module validations the rule violations would require correction. Using Rule Exclusion allows you to mark a Control Module violation as acceptable for a particular Rule(s) within a RuleSet for an Entity violation. Excluding a rule requires you to specify a reason for the exclusion, which can be viewed in the validation Results View when a validation has been executed. The Project Data View allows you to view and manage Rule exclusions for a configuration project.

#### **Excluding a Rule Violation**

When viewing the results of a Policy Validation you can choose to exclude an Entity Rule violation by selecting the Rule message item(s), right mouse-click in the results view to display its Context Menu and choose the menu item labeled **Exclude**. Only Rule message items that have a Warning or Error status can be excluded. A view will be displayed that requires you to specify an exclusion note. No exclusion is allowed without specifying a reason as to why the exclusion is being applied. Exclusion notes are saved in a Most Recently Used (MRU) list for fast retrieval of common reasons.





If a RuleSet that has violations reported on one or more Rule items has all its Rule violations excluded for the RuleSet, then an Audit (Success) type Rule message item is automatically added to the Results View as this RuleSet is now counted as a successful validation. Any removal of a Rule exclusion for a RuleSet that currently shows a successful validation through the added Success message will automatically remove this Audit (Success) we message.

#### Filtering the Results View to Hide Exclusions

When viewing the results of a Policy Validation you can choose to hide any Rule message items that are excluded by clicking the **Exclusions** toggle button. This button allows you to toggle the visibility of Rule message items that are currently excluded.

## Toggling Rule Exclusion Visibility in the Chart View

The Stacked Bar Chart View displays the results of the Policy Validation. You can toggle the visibility of whether excluded Rule message items are included in the Chart bar counts by toggling this button located near the top right of the Chart View.

#### **Removing Existing Rule Exclusions**

You can remove existing Rule exclusions by selecting the Rule message item(s), right mouse-click to display its Context Menu and select the menu item **Clear Exclusion**. This will permanently remove the exclusion status for the selected Rule message items.

Copyright © 2024 by Automation Software Technology, Inc. All Rights Reserved.

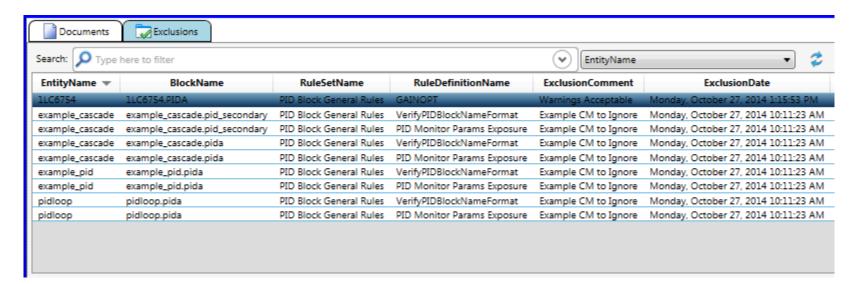
## Adding Rule Exclusions to a Result Set that Includes Multiple Projects

You can validate multiple versions of the same configuration project. When doing so, assure that the folder names where your configuration files reside contain similar formatting such as "VPS2\_20140621" and "VPS2\_20141001" where the common alias name by default is detected as the string value "VPS2". Rule exclusions are saved based on the alias name of the project derived from its folder path. This allows you to exclude an Entity name from a particular RuleSet Rule validation for all subsequent versions of the configuration project.

When comparing the results between multiple versions of the same configuration project, adding a Rule Exclusion for a particular Entity will result in all entities with the same name located in the Results View having this new Rule Exclusion applied.

### **Managing Rule Exclusions in the Project Data View**

You can view the current Rule exclusions for a particular Configuration Project using the Project Data View. This view allows you view and delete existing Rule exclusions.



#### Adding or Removing Rule Exclusions in Excel

The validation results displayed in the Results View can be copied using Ctrl+C to the Window Clipboard for pasting into Excel. You can modify the column data *IsExcluded* and *ExclusionNote* and then copy the Excel data to the Window Clipboard using Ctrl+C and click in the Results View and press Ctrl+V to paste the changes back into the Results View. Only changes made to the *IsExcluded* and *ExclusionNote* column data will be detected and updated. The exclusion date and time stamp will reflect the time that the data was pasted back into the Results View.

Empty ExclusionNote data values are not accepted and will not be updated in the Results View. You can remove and/or add Rule exclusions by editing the *IsExcluded* status in Excel. The *IsExcluded data column only accepts a boolean type value from the enumeration set {TRUE, FALSE}.* 

#### Rule Development Recommendations

When developing a RuleSet it is recommended that you only add a single success or "Audit" type message that indicates that all the rules contained in the RuleSet validated successfully. The Rule Exclusion functions expect that there will be only a single "Audit" type message that represents the status of all Rule validations. If you require additional information to be logged to the Results View when validating particular rules, then set the ResultState enumeration as "Info" only, which will categorize the result item as a "Message" type.

## **PolicyCop Rules**

#### **Rule Files**

AST provides a standard set of rules to use for the PKS Experion system that will validate policy for best practices and misuse of Honeywell Cxxx configuration.

To use these rules in the PolicyCop tool you need to specify where these rule files exist. Click the Browse Folder button located near the top of the PolicyCop View to specify the path where the rule files reside. When you change this file path, the rule files displayed in the Rule Sets list will be updated in the PolicyCop validation view. Any changes to the file contents in this folder path will automatically be detected and the existing Rule Files list will be automatically refreshed.

A Rule file contains a single Rule Set where a Rule Set defines a set of rule definitions. You will typically develop a Rule Set and its definitions to contain a specific validation requirement for a configuration object such as a Function Block type. There is no limit to the number of rule definitions contained in a Rule Set other than maintainability. You might organize Rule Sets based on whether they are general in nature or more specific that may not require periodic execution on a policy validation task. Rule definitions are executed beginning at their highest numerical priority level.

# **Creating/Modifying Rule Files**

You can create new rules or edit existing rules by clicking the web button located in the PolicyCop Validation View. Any changes to existing Rule files or the creation of new Rule files will automatically be reflected in the Rule Sets list in the PolicyCop Validation view.

AST is available to assist you in customizing existing Rule Sets or creating new for a service fee.

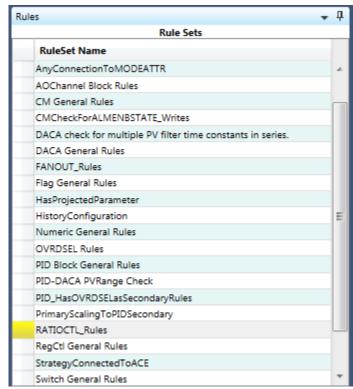
# **PolicyCop Rule Authoring**

Control Architect supports policy validation using the Workflow PolicyCop and CARuleAuthor that provides powerful rule authoring and execution on Honeywell PKS Experion Cxxx project configuration files. Policy rules are developed with a rules editor that can define:

- Parameter data values
- Function block attribute values
- Function block primary/secondary relationships
- Required function block dependencies
- Scaled value relationships
- Exposed pin connections
- Pin connection parameter references
- Template dependencies
- Best Practice Scenarios
- Function block design violations

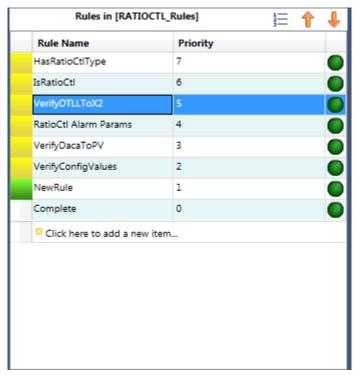
# **Rule Authoring View** Current RuleSet Folder Path 7 Load Configuration Asset Model data check 9 Load embedded Function blocks check 5 **3** 3 Rule Definition Methods **3** Rule Definitions for the selected RuleSet file 4 Code Snippets View **3** 21 Current Rule Definition Code Part Status Code Part View Expand/Collapse Button Auto-Set Rule Priority Button 12 iority 5 Reevaluation: Always Active ... Active Increase Rule Priority Button Decrease Rule Priority Button Find Results View 24

RuleSet Files found in the specified RuleSet Folder Path



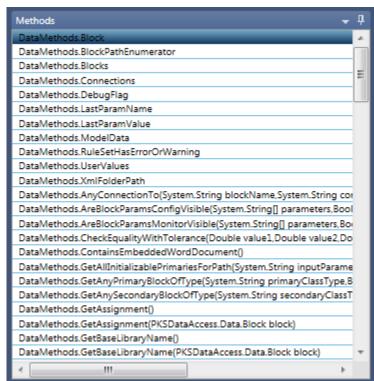
This is the list of available RuleSet files that have been found in the specified RuleSet folder path.

Rule Definitions for the selected RuleSet file



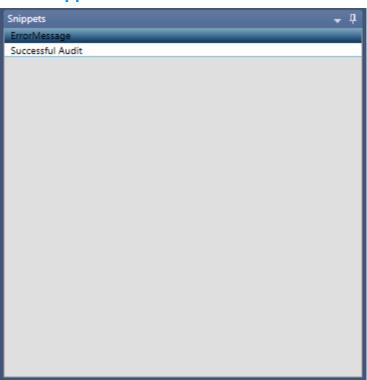
This is a list of Rule Definitions for the selected RuleSet.

### **Rule Definition Methods**



This is a list of available Rule methods and data objects for building Rule Definitions. You can hover over the item to view its ToolTip that describes its function. Left mouse-click and drag to drop the method or object in the code part editor.

#### **Code Snippets View**



Maintain a list of commonly used code snippets by dragging a selected code block to this view and dropping it onto its surface.



Load embedded Function blocks

Choose this option when data methods used are dependent on the embedded function blocks of a Module.

RuleSet Name

RATIOCTL\_Rules

Specify a RuleSet name that uniquely identifies the function of its contains Rule Definitions.

Current RuleSet Folder Path

C:\Software Development\ControlArchitect(WPF)\Rules Testing ▼

This is the current RuleSet folder path that has been selected.

RuleSet Description

RATIOCTL Usage Rules

Specify a description for this selected RuleSet. This description is presented in the results view when policy validation is executed.

Load Configuration Asset Model data check

▼ Load Xml Model data

Choose this option when data methods used are dependent on the Configuration Asset data Model.

RuleSet Folder Browse Button



Click this button to browse for the folder where your RuleSet files will reside.

RuleSet Chaining Behavior

RuleChaining Behavior: None ▼

Set the RuleSet chaining behavior. This is normally set to the value [None] unless special rules are being written which require reevaluation such as when enumerating collection items.

Auto-Set Rule Priority Button

1 2 3

Click this button to auto-set the priority for all Rule Definitions in the RuleSet.

Increase Rule Priority Button



Click this button to move the selected Rule Definition to a higher processing order in the list.

Decrease Rule Priority Button



Click this button to move the selected Rule Definition to a lower processing order in the list.

→ WordWrap check



Choose this option to force Code part expressions to wrap within the current editor view size.

Priority



Specify the execution priority for this Rule Definition. A Rule Definition with a priority of 0 is the lowest priority. Rule Definitions are executed in descending priority order, which is top-down in the list. Click the Auto-Set Priority button to automatically set the priorities based on the Rule Definition location in the list where the last item has a rule priority of 0.

Description

Verify RatioCtl Has Dependency Blocks

Specify a description for this Rule Definition.



#### Reevaluation



Choose the Reevaluation behavior for the Rule Definition.



#### **Active**



Check this to make this Rule Definition active during rule execution. Removing this check will make this Rule Definition inactive and ignored during rule execution.



## **Code Part View Expand/Collapse Button**



Click this button to expand/collapse the focused code part view. Use this to expand the view surface when expressions become large and/or complex. Click again to collapse the view to its normal size.

21

## **Current Rule Definition Code Part Status**



This image indicates the expression parse validation status when changes are made to expressions in the code part.

22

#### **Snippet Editor**



This editor allows you to edit code snippets that have been auto-generated by dragging a code selection block to the Snippet View.

23

### **Edit Status**



Row header color indicates edit status, such as new, modified, and error. Hover with mouse cursor to view ToolTip of status information.

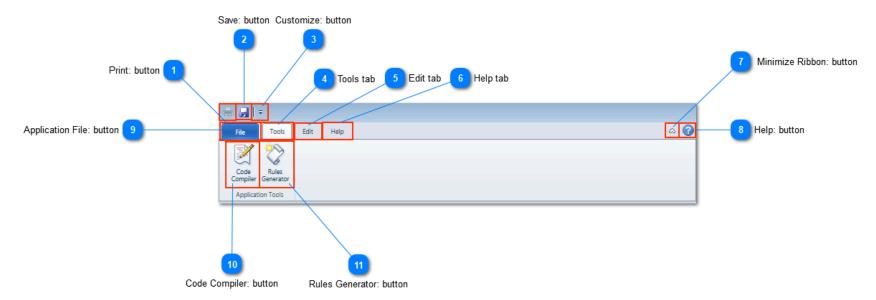
24

## Find Results View



Click this tab to view results of a Find Text operation.

#### **Ribbon Tools tab**







Click this button to print the contents of the active Rule set.

Save: button



Click this button to save changes to any Rule definitions.

Customize: button



Click this drop-down button to customize the Ribbon Toolbar.

Tools tab



Click this tab to access the Tools features.

5 Edit tab



Click this tab to access the Edit features.

6 Help tab

Click this tab to access the Help features.

Minimize Ribbon: button



Click this button to minimize/maximize the Ribbon Toolbar space.

Help: button



Click this button to view this help topic.

Application File: button



Code Compiler: button



Click this button to launch the Code Editor used to author user defined methods that can be referenced in Policy Rule definitions.

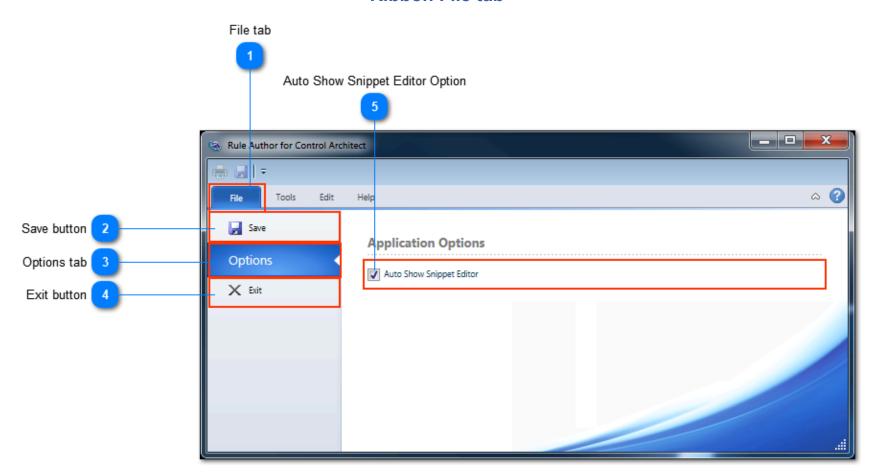


# **Rules Generator: button**



Click this button to display the rules Auto-Generate view. This feature allows you to automatically generate rules for visual blocks that have been selected in a Chart View.

#### Ribbon File tab





Click this tab to view File options.

Save button

Click this button to save all modifications.

Options tab
Options

Click this tab to view Application options.

Exit button

× Exit

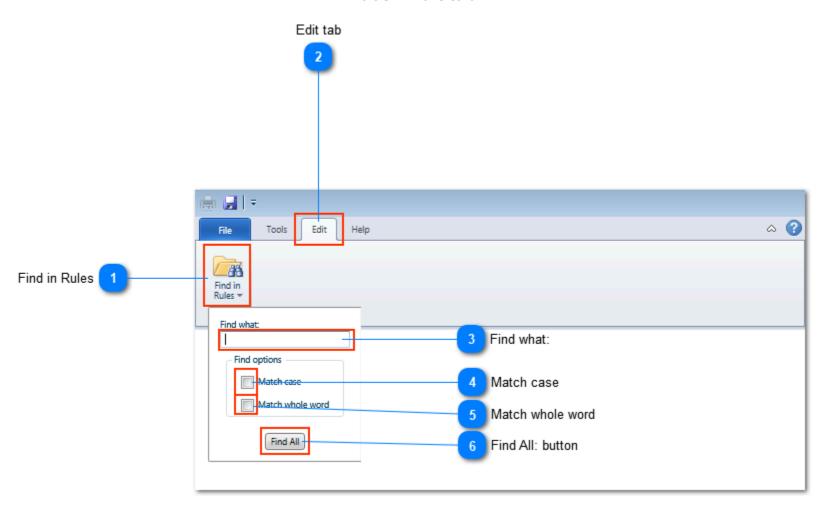
Click this button to close the application.

Auto Show Snippet Editor Option

Auto Show Snippet Editor

Choose this option to automatically bring into view and focus the Snippet Edit View when a User Snippet is selected in the list of User Snippets.

#### **Ribbon Edit tab**



Find in Rules



Click the drop-down to specify search criteria to find text in all RuleSet rules.

2 Edit tab

Find what:

Specify the text to search for in all Rule Sets.

Match case

Choose this option to search for text using the specified character case.

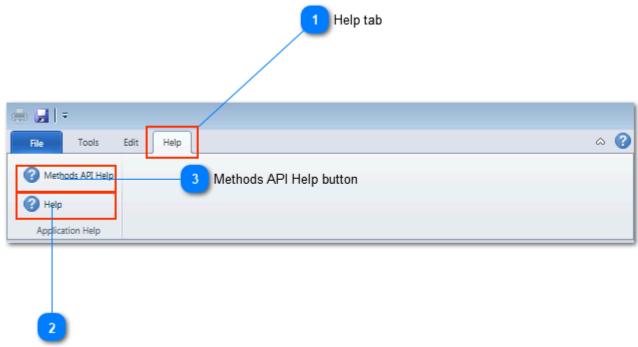
Match whole word

Choose this option to search for text found as whole words.

Find All: button

Click this button to begin searching all Rule Sets and their rules for the specified text.

# Ribbon Help tab



Application Help button



This Ribbon Tab provides access to Help documentation.

Application Help button

Click this button to view the Help document for this Rule Authoring application.

Methods API Help button

Methods API Help

Click this button to view the Help document for the Methods API.

#### **Auto-Rule Generation**

The rules Editor supports auto-rule generation based on Function blocks currently selected in a ChartView. Communications between the Control Architect running instance and the Rule Editor application is accomplished using Microsoft Messaging Queuing Service.

# **Microsoft Messaging Queuing Service**

This service is enabled and configured during the installation of Control Architect. If messages related to failure to access the Microsoft Messaging Queuing Service are received perform the following activities to enable the service.

#### To install Message Queuing 4.0 on Windows 7

- 1. Open Control Panel.
- 2. Click Programs and then, under Programs and Features, click Turn Windows Features on and off.
- 3. Expand Microsoft Message Queue (MSMQ) Server, expand Microsoft Message Queue (MSMQ) Server Core, and then select the check boxes for the following Message Queuing features to install:
  - a. MSMQ HTTP Support.
- 4. Click OK.
- 5. If you are prompted to restart the computer, click OK to complete the installation.

#### **Tips and Tricks**

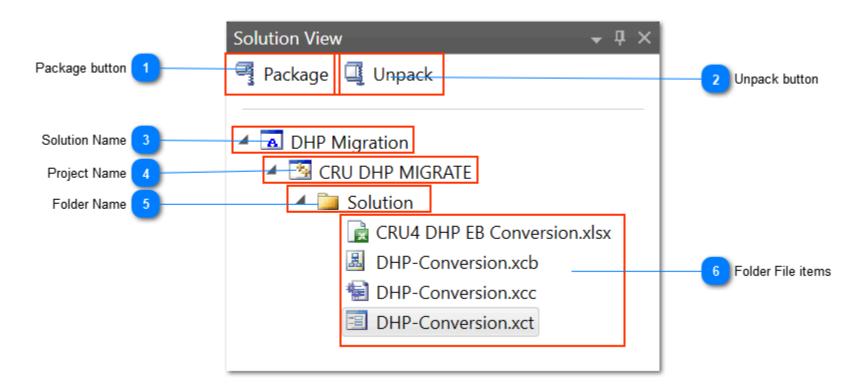
If the Messaging Queue fails during startup of the Rule Authoring application verify that the existing queue has permissions for the User Account the application is running as.

You can view the created message queue through the Computer Management console. Expand *Message Queuing\Private Queues* to see the application created queue named *SelectionChanges*. Right mouse-click on this queue to view its properties and verify access permissions on the Security tab.

#### **Solution View**

The Solution view provides a hierarchical structure for maintaining projects, template instances, and data files associated with Control Conversion tasks. This is a licensed option. With this Solution view you are able to define Control Conversion task templates that facilitate Bulk

Build/Edit operations on Honeywell supported configuration file types including Cxxx (XML), PKS Scada (PNT) and LCN (EB).



# Package button



Click this button to package up one or more projects in your Workspace Solution. All files referenced in the Project including Excel, Notes, etc. will be packaged into a compressed file for deployment to other systems or simply as a backup.

# Unpack button



Click this button to unpack a previously packaged up Solution project(s). Users can choose to unpack while maintaining the original folder and file hierarchy or choose to unpack to a flat folder.

# Solution Name

DHP Migration

This is the Solution Name item. This typically is also the name of the Workspace file. A workspace file can only contain a single Solution item.

# Project Name

▲ CRU DHP MIGRATE

This is a Project Name item. Workspace solutions can contain multiple project items to help organize the solution. Typically a Project will contain the solution files required to convert a hardware type such as DHP conversion to Scada or PIU conversion to C300, etc.

# Folder Name

■ Solution

This is a Folder Name item. Projects will contain one or more Folder items that represent folder paths in your file system. A folder item is used to organize files related to a specific Control Conversion task. You can choose to add other file types used in the task such as your Excel document, notes, image captures, etc. These other file types can be quickly opened up in their respective application by left mouse double-click.

# Folder File items

☐ CRU4 DHP EB Conversion.xlsx☐ DHP-Conversion.xcb

DHP-Conversion.xct

These file items typically reference documents that are stored in the file system folder path specified in the project Folder item. You can optionally add a reference file here that is outside the folder path.

- \*.XCT is always associated with a Control Property Template file that defines either source (Excel) and/or target (Control Configuration) properties that will be bound in your Control Binding Template file.
- \*.XCB is always associated with a Control Binding Template file that defines data mappings between your source and target properties with optional transformation script written in C# language.
- \*.XCC is always associated with a Control Conversion Template file that allows you to bind a Control Binding Template to an Excel Data Source file.

Other file types can be added to the folder item in your Solution view to help organize and provide for quick open in its native application type such as Excel, Image, and Word type documents. Items can be quickly viewed using the left mouse double-click action.

# **Using the Solution View**

# **Solution View Organization**

The Solution View supports Control Architect functions such as Control Conversion. Control Conversion is basically defined as taking a source of data such as an Excel spreadsheet file, converting the data values, and exporting the converted data files to a known data format such as a Honeywell Exception Build (EB) file or a Honeywell Cxxx XML configuration file.

Control Architect provides a set of Templates that are used to facilitate a Control Conversion task. When adding a Template to a Solution View project data folder, a copy of this template is created and becomes available as a file.

#### **Create a New Solution**

To create a new Solution, choose the main menu item File-New Solution.

#### Add a New Project

A Solution can only contain Projects as direct subitems. You <u>cannot</u> add a Project as a subitem to another Project.

To add a new Project to the Solution, right mouse-click on the Solution Name item in the view to display its context menu and choose the menu item **Add-New Project**...

# Add a Folder to the Project

A Project can only contain Folders as direct subitems.

To add a new data Folder to a Project, right mouse-click on a Project item in the Solution View to display its context menu and choose the menu item **Add-New Folder**. A folder browse dialog view will appear to allow you to choose a file system Folder to add as a data folder.

#### Add Existing file items to the Project Data Folder

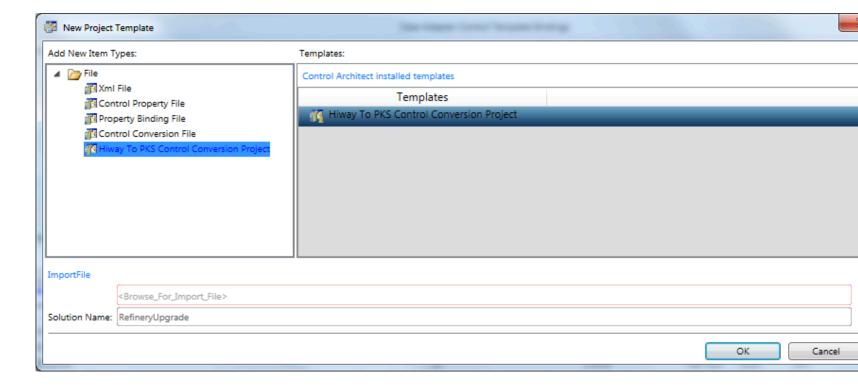
File items can only be contained in a Project Data Folder.

To add an existing file item to a Project data folder, right mouse-click on a Project item in the Solution View to display its context menu and choose the menu item **Add-Existing Item**... A file browse dialog view will appear to allow you to choose files to add to the data folder view.

You can add any type of file to the project data folder. Select multiple items to add them in the same operation.

#### Add New items to the Project Data Folder

To add an existing file item to a Project data folder, right mouse-click on a Project item in the Solution View to display its context menu and choose the menu item **Add-New Item**... A template browse dialog view will appear to allow you to choose from a set of Control Architect installed templates to add to the data folder view.

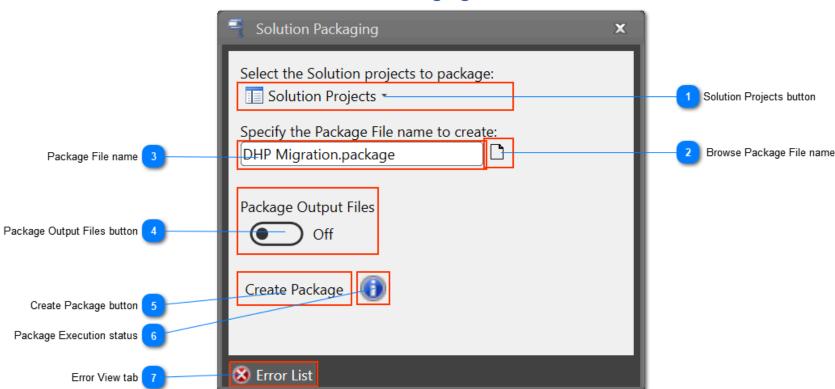


### Working with the View Items

- You can move file items between data folders by left mouse-clicking on the item and dragging it to another data folder. This will move the physical location of the file item being dragged to the new file system location.
- You can copy file items between data folders by left mouse-clicking on the item and dragging it to another data folder while pressing the **Ctrl** key. This will copy the physical location of the file item being dragged to the new file system location. If the file item already exists in the destination file folder or if a name of a file item currently exists in the destination Solution Project folder, then the copy operation will fail.

- You can copy an existing Project Data Folder or Project File item within a data folder by right mouse-clicking to display a context menu and choosing the menu item *Copy*. This will create a new folder in the file system and name it by appending the text "\_*Copy*". All file items will also be copied to the new folder location and renamed by appending the text "\_*Copy*".
- You can remove a file item from a Project Data folder by right mouse-clicking to display a context menu and choosing the menu item *Remove*. You will be prompted to remove the file item from the Solution View and/or permanently delete the file item.
- You can left mouse double-click on a file item to execute the verb registered on your system for the specific file type. For example if you were to add an Excel Workbook file reference and double-click on the Workbook file item it would launch Excel with the opened Workbook since the verb "Open" is default registered on your system.

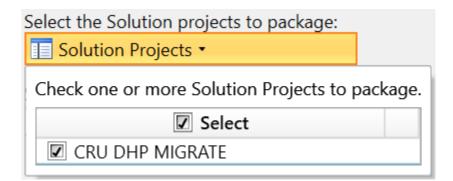
## **Solution Packaging window**



# Solution Projects button

■ Solution Projects \*

Click the drop-down button to view a list of projects that can be selected for packaging. By default all projects contained in the active solution are selected for packaging.



# Browse Package File name

Click this button to browse for a file location to create the new Package file. By default a file name using the extension .package will be created.

# Package File name

DHP Migration.package

Specifies the path and name of the new Package file to create.

# Package Output Files button

Package Output Files



Choose this option to package up build content that was generated to the output folder specified in the Control Conversion template.

# Create Package button

Create Package

Click this button to begin the package execution.



# **Package Execution status**



Indicates the status of the package execution.

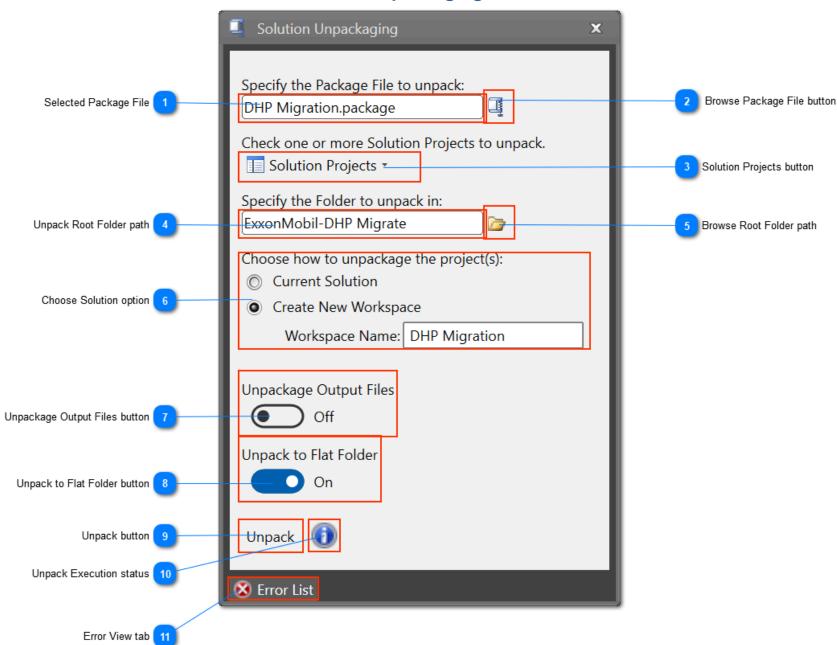


### **Error View tab**



Click this tab to view any errors encountered during the package execution. This tab will flash red if any errors were encountered.

## **Solution Unpackaging window**



Selected Package File

DHP Migration.package

Specifies the Package file that has been selected for unpacking of one or more projects.

Browse Package File button



Click this button to browse for a \*.package file to unpack. Only files that have been created using the Package view are supported.

Solution Projects button



Click the drop-down buttion to select which projects in the package file are to be unpacked. By default all projects contained in the package file are selected for unpacking.

Unpack Root Folder path

ExxonMobil-DHP Migrate

Specifies the root folder path that has been selected to contain the unpacked project files.

Browse Root Folder path



Click this button to select the folder in your file system where the project file items will be unpacked into. Users can choose to unpack to this root folder path while maintaining the original folder hierarchy or choose to unpack to this single folder path.

# 6 Choose Solution option

Choose how to unpackage the project(s):

Current Solution

Create New Workspace

Workspace Name: DHP Migration

Choose to unpack the projects contained in the selected Package file to the active Solution Workspace or choose to create a new Workspace file to contain this new Solution and project(s).

You cannot unpack a project into your active Solution that has the same name.

# Unpackage Output Files button

Unpackage Output Files



)ff

Choose this option to unpack optional output files. Output files are those files that were generated in a Control Conversion task and optionally packaged up.

# Unpack to Flat Folder button

Unpack to Flat Folder



On

Choose this option to unpack all package content to the selected Root folder path without preserving the original folder hierarchy.

# Unpack button

Unpack

Click this button to begin unpacking the selected projects contained in the package file.

#### Unpack Execution status



Indicates the status of the unpack execution.

#### Error View tab



Click this tab to view any errors encountered during the unpack execution. This tab will flash red if any errors were encountered.

### **Installed project item Templates**

#### **Overview**

Control Architect provides a set of Templates to facilitate <u>Control Conversion</u> type tasks. Control Conversion is basically defined as taking a source of data such as an Excel Spreadsheet file, converting the data values, and exporting the converted data files to a known data format such as a Honeywell Exception Build (EB) file or a Honeywell Cxxx XML configuration file.

The current release of Control Architect supports conversion of source data located in an Excel file to Honeywell C300 XML based configuration files for importing control strategies into an EPKS Server Engineering Repository Database (ERDB). Typically this data will represent an engineering database or Hiway based configuration that has been parsed from Honeywell Exception Build (EB) files to an Excel file Data Source. Control Architect supports parsing of EB files to Excel using a Workflow based task execution.

#### **Control conversion templates**

Control Architect provides a set of templates that provide configurable views for defining a Control conversion task.

#### **Control Property Template**

The **Control Property Template** provides a view for defining Control properties or parameters.

#### **Control Binding Template**

The <u>Control Binding Template</u> provides a view for defining property binding relationships or mappings between source and target Control properties. Binding transformation expressions written in C# language script are also supported to define how a source property value is converted to its target property value.

#### **Control Conversion Template**

The <u>Control Conversion Template</u> provides a view for defining Control conversion tasks. A Control conversion task allows you to attach an Excel file as a Data Source to a Control Binding template. This Control conversion task when executed will generate a Data Collection view that specifies how Control strategies and their properties (parameters) will be updated. A Workflow task can be defined that will attach to this generated Data Collection view and provide the actual <u>Bulk Edit/Build</u> operations.

#### **Hiway Control Conversion to EPKS Project Template**

This solution for converting Hiway data point configuration to PKS CEE resident Control strategies provides predefined sets of Control Conversion templates that when executed against a Hiway EB based Excel Data Source will convert all its contained Hiway Regulatory Box algorithms and PIU data point configurations. This project solution is made available from AST and can be downloaded from its website upon request.

This solution is packaged into a Zip file that is referenced as the Import file to add as a new item to a Project folder in the Solution View.

## **Bulk Building/Editing**

Control Architect supports bulk building and editing of existing Cxxx and ACE based Control modules. Predefined Control Conversion templates and their views support the tasks required to define source and target properties, their binding (mapping) relationships, and the adapter binding to a Data Source that contains the Microsoft Excel based data to use in the build or edit Control Conversion tasks.

This is an optional licensed component within Control Architect.

#### **A Control Conversion Workflow Process**

A Control Conversion Workflow process requires 4 Control Conversion Template instances to be defined and a Data Collection to be executed on the Control Conversion template itself. The generated Data Collection is then referenced in a Workflow Task, which is executed to perform the building or editing of the selected output type.

- 1. Add an instance of a Control Property Template and specify the properties that define the source of the conversion data. This would be data that resides in a Microsoft Excel Workbook file.
- 2. Add another instance of a Control Property Template and specify the properties that define the target of the conversion process. This would typically be Cxxx configuration parameter names.
- 3. Add an instance of a Control Binding Template to specify the binding relationships between your defined source and target property definitions. Expressions may need to be developed to transform the property values from the source into valid values required for the target Control System platform.
- 4. Add an instance of a Control Conversion Template to specify the Data Source mapping and the output adapter to use. Currently Control Architect Control Conversion functions support XPM, PKS Cxxx/ACE, SCADA, and LCN EB output adapters.
- 5. Execute a Data Collection generation from within the Control Conversion Template view.
- 6. Add a Control Conversion Task to this Workflow view specifying the type of *ControlConversion Activity* to execute on the saved Data Collection data.

#### Tip

To quickly generate a Control Conversion project use the **Create Control Conversion Project Workflow**.

### **Using the Control Conversion Workflow Manager Task View**

#### **Workflow Session**

A Workflow session allows you to define one or more Control Conversion tasks to be executed. Tasks defined in the Workflow session are saved to an external file and auto-restored when this Task view is opened.

Only a single Task view can be opened at the same time.

#### **Honeywell EB File to Excel Conversion Workflow Activity**

This conversion activity [EBToExcelActivity] supports converting Honeywell Exception Build (EB) type file data contained in text files to a Microsoft Excel Workbook file. You can convert one or more EB files during the same task execution to the same Excel Workbook where the Spreadsheet tabs represent a specific Template or point type found during the EB file parsing.

#### **PKS Control Conversion Workflow Activity**

This conversion activity [*PKSConvertActivity*] supports converting property data specified in a Data Collection to PKS Cxxx/ACE XML Configuration files. The Data Collection is generated from within the Control Conversion Template view. Options within the Data Collection view allow you to enable/disable entities and their property data that has been detected as changed (Bulk Edit) or requiring building (Bulk Build).

#### **SCADA Control Conversion Workflow Activity**

This conversion activity [SCADAConvertActivity] supports converting property data specified in a Data Collection to Quick Builder PNT Configuration text files. The Data Collection is generated from within the Control Conversion Template view. Options within the Data Collection view allow you to enable/disable entities and their property data that has been detected as changed (Bulk Edit) or requiring building (Bulk Build). The generated PNT files are imported into Quick Builder using its Import function.

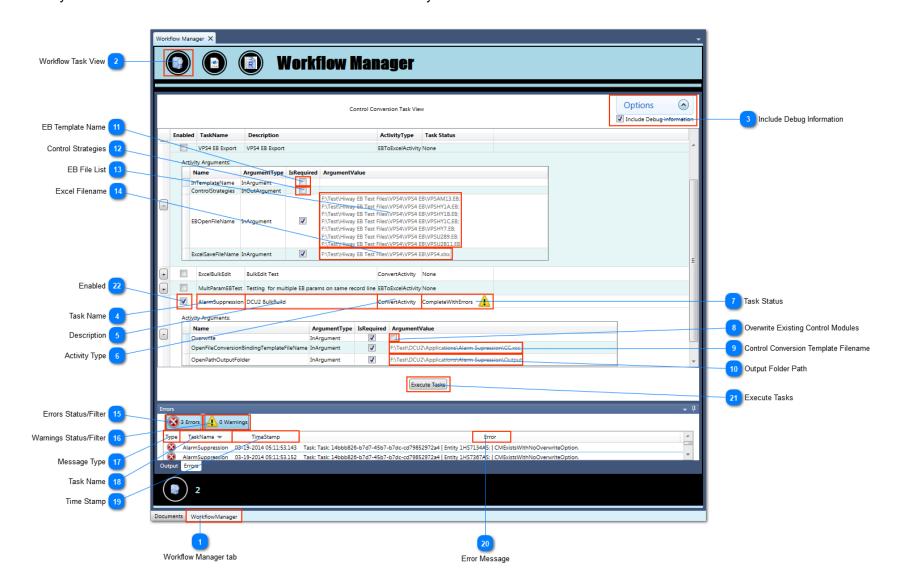
#### **EB Control Conversion Workflow Activity**

This conversion activity [EBConvertActivity] supports converting property data specified in a Data Collection to LCN Exception Build (EB) Configuration text files. The Data Collection is generated from within the Control Conversion Template view. Options within the Data Collection view allow you to enable/disable entities and their property data that has been detected as changed (Bulk Edit) or requiring building (Bulk Build). The generated EB files are imported into the LCN using the PED Exception Build function.

#### **Task Execution Results**

#### **Workflow Task View**

This is an example of a Workflow Task View where multiple tasks have been defined. This example shows the configuration for an EB-toExcel activity based Conversion task and also a Control Conversion activity based task.



Workflow Manager tab

WorkflowManager

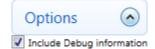
This application tab contains the Workflow Manager view.

Workflow Task View



Click this button to add a Workflow Task view. You can have multiple Task views opened at the same time, however the last view closed will be the view session that will be saved and auto-restored during your next view session.

Include Debug Information



Choose this option to report Debug type information during a task execution.

Task Name

AlarmSuppression

Specify a name for this Task.

Description

DCU2 BulkBuild

Specify a description for this Task.

6 Activity Type

ConvertActivity

Select an Activity type from the drop-down list of currently supported Control Conversion activities.

Copyright © 2024 by Automation Software Technology, Inc. All Rights Reserved.

7 Task Status
CompleteWithErrors

The current status of the Task. This status will display a Progress bar when the task is executing.

# Overwrite Existing Control Modules

Choose this option to allow overwriting existing Control Module configuration files residing the specified output folder path.

Control Conversion Template Filename

F:\Test\DCU2\Applications\Alarm Supression\CC.xcc

Select the Control Conversion Template Filename that was used to generate a Data Collection that will be used in this Control Conversion task.

Output Folder Path

F:\Test\DCU2\Applications\Alarm Supression\Output

Specify the output folder path where your new Cxxx Control Modules will be generated or edited.

EB Template Name

Optionally specify a valid point type Template name to convert to Excel.

Control Strategies

This if for internal use only.

EB File List

F:\Test\Hiway EB Test Files\VPS4\VPS4 EB\VPSAM13.EB;
F:\Test\Hiway EB Test Files\VPS4\VPS4 EB\VPSHY1A.EB;
F:\Test\Hiway EB Test Files\VPS4\VPS4 EB\VPSHY1B.EB;
F:\Test\Hiway EB Test Files\VPS4\VPS4 EB\VPSHY1C.EB;
F:\Test\Hiway EB Test Files\VPS4\VPS4 EB\VPSHY7.EB;
F:\Test\Hiway EB Test Files\VPS4\VPS4 EB\VPSU2B9.EB;
F:\Test\Hiway EB Test Files\VPS4\VPS4 EB\VPSU2B1.EB

Specify or select from the Open File view the list of Honeywell Exception Build (EB) files to convert to Excel.

**Excel Filename** 

F:\Test\Hiway EB Test Files\VPS4\VPS4 EB\VPS4.xlsx

Specify or select from the Save File view the Excel filename to be created.

Errors Status/Filter

3 Errors

This toggle button indicates the current number of error type messages contained in the [Errors] tab. Click this button to toggle the visibility of Error type messages in the [Errors] tab.

**16** Warnings Status/Filter



This toggle button indicates the current number of warning type messages contained in the [Errors] tab. Click this button to toggle the visibility of Warning type messages in the [Errors] tab.

Message Type

Type

Indicates the type of message.

Task Name

TaskName

Indicates the Task name that sent this message during execution.

19 Time Stamp

TimeStamp

Indicates the time of execution that this message was sent.



Indicates the contents of the message.

**Execute Tasks** 

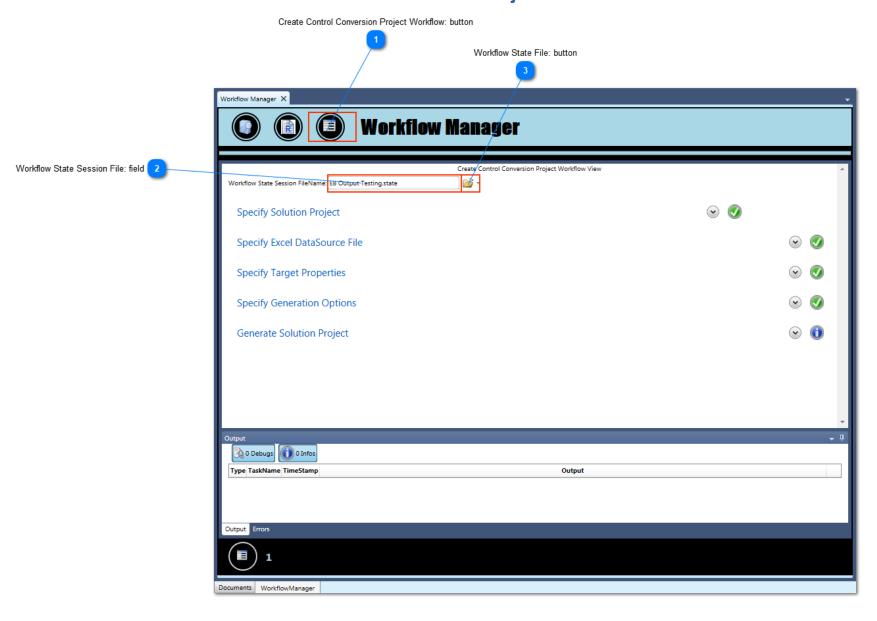
Execute Tasks

Click this button to begin executing enabled conversion tasks.

Enabled

Check this to enable the task for execution. Multiple tasks can be executed in the same operation. Multiple tasks are executed in series.

## **Create Control Conversion Project Workflow**



Create Control Conversion Project Workflow: button



Click this button to display this Create Control Conversion Project Workflow view.

Workflow State Session File: field

EB Output Testing.state

Displays the selected Workflow State Session file path. Click the drop-down button to open, save, and restore a Workflow state to the view contents.

Norkflow State File: button

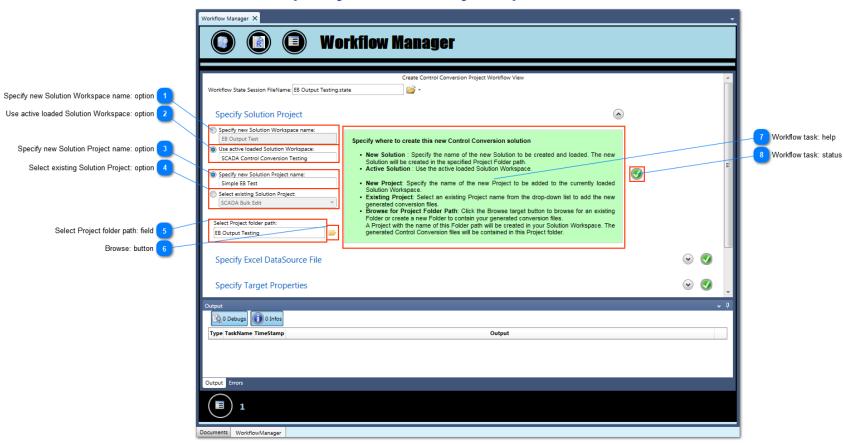


Click the drop-down button to open, save, and restore the Workflow state session for later recall.

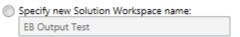
The data specified in this Workflow can be saved to a State Session file so that it can quickly be restored when returning to this Workflow view. The last used Workflow State Session file is automatically referenced, but you must click the drop-down button **Restore** menu function to restore the data contained in the State Session file to the Workflow view content.

A previously saved Workflow State Session file can be used to develop another type of Control Conversion Project by opening/restoring a previously saved Session file, changing its contents and saving the new Workflow Session data to a new Session State file.

#### **Specify Solution Project options**

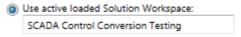


Specify new Solution Workspace name: option



Choose this option to create a new Solution Workspace to contain the generated Project files.

Use active loaded Solution Workspace: option



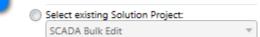
Choose this option to generate the project files to the currently loaded Solution Workspace.

Specify new Solution Project name: option



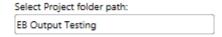
Choose this option to specify a new Project name that will be created in the Solution Workspace.

Select existing Solution Project: option



Choose this option to select an existing Project in the currently loaded Solution Workspace where the generated Control Conversion files will reside.

Select Project folder path: field



Displays the folder path where the generated project files will reside. Click the Browse button to specify the folder path.

Browse: button



Click this button to specify the folder path where the generated project files will reside.

Workflow task: help

#### Specify where to create this new Control Conversion solution

- . New Solution : Specify the name of the new Solution to be created and loaded. The new Solution will be created in the specified Project Folder path.
- Active Solution: Use the active loaded Solution Workspace.
- . New Project: Specify the name of the new Project to be added to the currently loaded
- · Existing Project: Select an existing Project name from the drop-down list to add the new
- generated conversion files.

   Browse for Project Folder Path: Click the Browse target button to browse for an existing Folder or create a new Folder to contain your generated conversion files. A Project with the name of this Folder path will be created in your Solution Workspace. The generated Control Conversion files will be contained in this Project folder.

Provides quick help tips for completing this task. The background color of this quick help view will change to indicate its completion status.

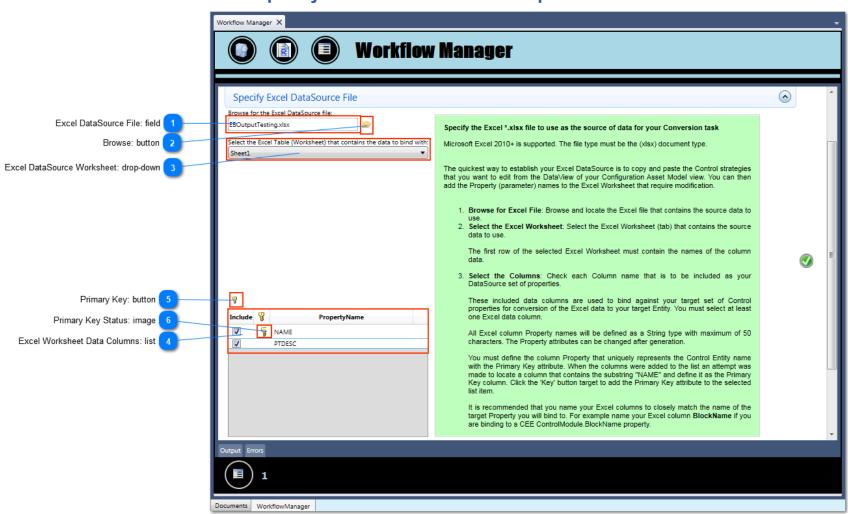


## **Workflow task: status**



Provides status for this task completion. Any error or warnings encountered in the data for this Specify Solution Project task will result in a Status change.

## **Specify Excel DataSource File options**



# **Excel DataSource File: field**

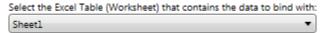
EBOutputTesting.xlsx

Displays the Excel DataSource file path. Click the Browse button to select the Excel file to use as the source of the data to use for building or editing Control strategies.

# Browse: button

Click this button to select the Excel file to use as the source of the data to use for building or editing Control strategies.

## Excel DataSource Worksheet: drop-down



Select the Excel Worksheet name to use as the source of the data.

#### Excel Worksheet Data Columns: list



This list contains the Excel data columns that are discovered in the selected Excel Worksheet. When an Excel Worksheet is selected from the Excel Worksheet drop-down, this list will be populated with data column names discovered in the Worksheet.

When your Control Conversion project files are created, this set of source properties is used to create the <u>Control Property Template</u> file.

## Primary Key: button

P

Click this button to specify the selected Excel data column as the source of the data that uniquely identifies the data contained in this Excel Worksheet. This will typically be the data column that specifies the name of the new Control Strategy or Control Module.



There must be a *Primary Key* defined for a single Excel data column. Failure to specify a Primary Key will result in your Data Collection execution failing.

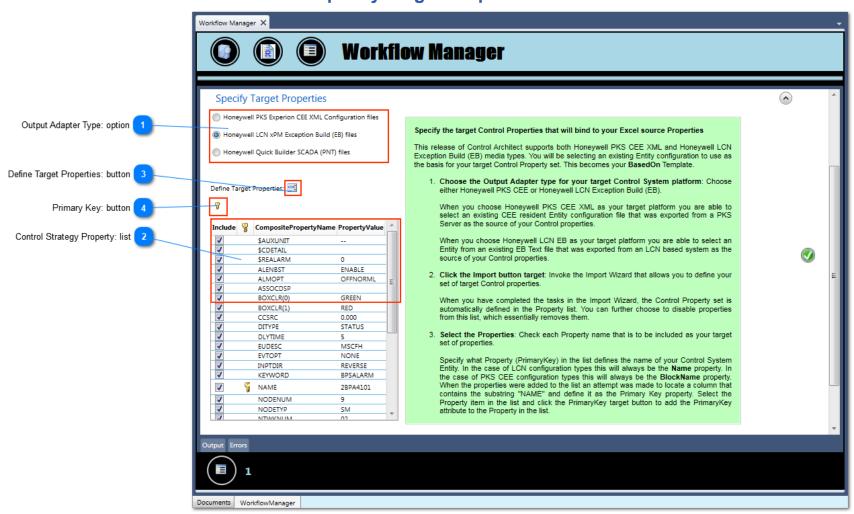


## **Primary Key Status: image**



This image identifies the Excel data column that uniquely identifies all data contained in the selected Excel Worksheet. Click the **Primary Key button** to toggle this status for the selected Excel data column in this list.

## **Specify Target Properties**



## Output Adapter Type: option

- Honeywell PKS Experion CEE XML Configuration files
- Honeywell LCN xPM Exception Build (EB) files
- Honeywell Quick Builder SCADA (PNT) files

Choose the target Control System Platform type to use for selection of the Output Adapter used during Control Conversion execution. This release supports the building and/or editing of PKS Cxxx/ACE, Quick Builder SCADA, and LCN Exception Build (EB) Control strategies.

## **Control Strategy Property: list**

Include	B	CompositePropertyName	PropertyValue	٨
1		\$AUXUNIT		
<b>J</b>		\$CDETAIL		
J		\$REALARM	0	
J		ALENBST	ENABLE	
<b>J</b>		ALMOPT	OFFNORML	=
<b>J</b>		ASSOCDSP		П
1		BOXCLR(0)	GREEN	

This list contains the target Control Strategy properties that were defined using the Import Wizard.

When your Control Conversion project files are created, this set of target properties is used to create the Control Property Template file. A Control Binding Template file will also be created that binds your Excel DataSource columns to properties defined in this list. This Workflow will attempt to automatically bind same named Excel columns and Control Strategy properties. If you choose to name data columns in your Excel Worksheet that do not fit the expected target Control Strategy properties, then you may have to manually establish the bindings in the generated Control Binding Template file.

#### **Define Target Properties: button**



Click this button to invoke the Import Wizard that allows you to choose the Control Strategy that will be used to define the target Control Template properties. This Import Wizard allows you to select an existing PKS XML CM Configuration file, an Entity from an EB file, an Entity from a SCADA PNT export file to use as the basis for defining your target Control properties. The configuration entity that is chosen here is then defined as the BasedOn Template that is subsequently used to build or edit Control strategies.

When your Control Conversion project files are created, this set of target properties is used to create the Control Property Template file.

# 4

#### **Primary Key: button**

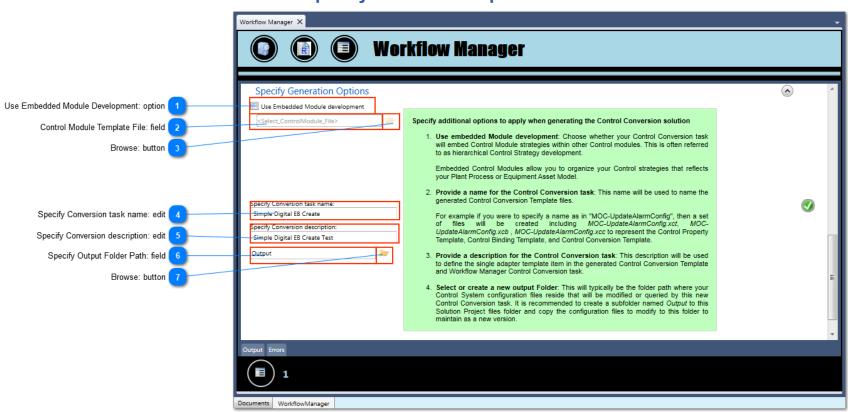


Click this button to specify the selected Control Property as the source of the data that uniquely identifies the data contained in this Control Strategy. This will typically be Property that specifies the name or BlockName of the new Control Strategy or Control Module.



There must be a *Primary Key* defined for a single Control Property. Failure to specify a Primary Key will result in your Data Collection execution failing.

## **Specify Generation options**



## Use Embedded Module Development: option

Use Embedded Module development

Choose this option to specify in your generated <u>Control Conversion Template</u> to use embedded Control Module development. This option only applies to the PKS Control Module output type. When choosing this option specify the Control Module XML configuration file that will be used as the typical or BasedOn Template.

# Control Module Template File: field

<Select\_ControlModule\_File>

Displays the XML Configuration file path for the selected Control Module to use as the Template or BasedOn source when creating container Control modules.

## Browse: button

Click this button to browse for the Control Module XML configuration file that will be used as the source of the Template or BasedOn file when creating containing Control modules.

# Specify Conversion task name: edit

Specify Conversion task name: Simple Digital EB Create

Specify the name of this conversion task that will be used to name the generated Control Conversion files. In this example where "Simple Digital EB Create" is specified the following files will be generated:

- 1. [Simple Digital EB Create.xct] Control Property Template file
- 2. [Simple Digital EB Create.xcb] Control Binding Template file
- 3. [Simple Digital EB Create.xcc] Control Conversion Template file

## Specify Conversion description: edit

Specify Conversion description: Simple Digital EB Create Test

Specify a description that identifies this Control Conversion project. This description will be used when adding a new Control Conversion Activity task to the <u>Workflow Manager Control Conversion view</u>.

# Specify Output Folder Path: field

Output

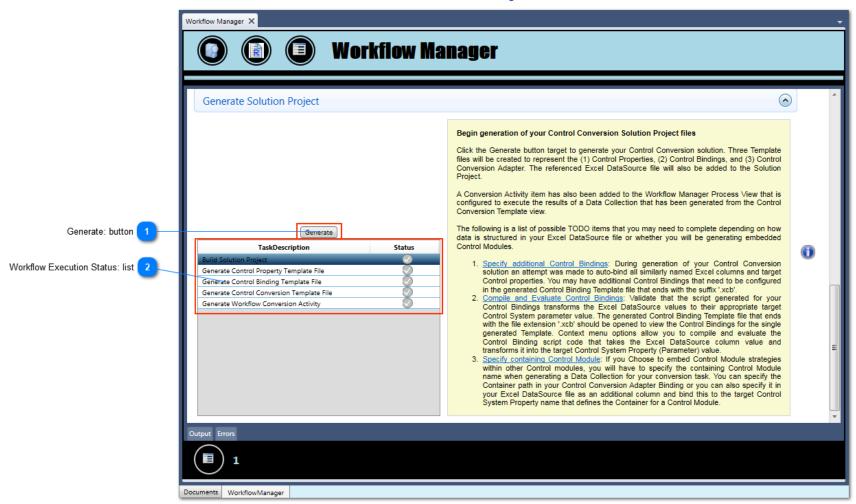
Displays the selected output folder path. This folder path is where your configuration output files are edited or created when your Control Conversion task is <u>executed</u> for this project. Typically you will create a folder named "Output" that is a subfolder to the folder that contains your generated project template files.

# Browse: button



Click this button to browse or create a folder path to contain the output files as a result of executing this project in a Control Conversion task. When executing a <u>Data Collection</u> from the Control Conversion Template view this folder path will be referenced to identify editing or creation of new Control strategies in the generated Data Collection.

#### **Generate Solution Project**

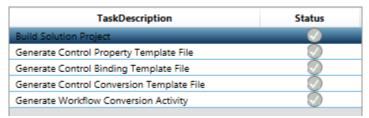


Generate: button

Generate

Click this button to generate your Control Conversion project based on the data provided in this Workflow.

#### Workflow Execution Status: list



Displays the status of the Control Conversion project generation.

If any errors or warnings are encountered during the generation, they can be viewed in the Errors tab. Debug and descriptive output information can be viewed in the Output tab.



This Workflow execution cannot overwrite or edit existing same named project files. If an execution discovers that a project file already exists with the same name it will fail and identify this error. The specific Workflow task section noted in this list will change its Status image to reflect the failure type.

To reuse this Workflow to change the way existing Project files were configured, you must first remove them from the Solution Project. This can be done by right, mouse-clicking on the Project item in the Solution Workspace Project to display its shortcut menu and choosing the menu item "**Remove**". Choose the prompted Message Box option *Delete* to permanently delete this file.

#### **Control Conversion views**

Control Conversion is a set of application views and functions that allows users to develop Entity/Parameter conversion tasks between an Excel DataSource and a Control System Process Control Strategy.

This set of functions is intended for engineers that have a need to migrate legacy Honeywell or other vendor Control System platforms to Honeywell based Control systems. AST will provide preconfigured templates that support Honeywell Hiway migration to PKS Experion CEE nodes.

The current release of Control Architect supports PKS Cxxx/ACE type configuration as the output adapter for a Control Conversion task. Future releases of Control Architect will support the Honeywell UCN using an EB File as the output adapter and also integrate with Honeywell TPS Builder through a TPSB output adapter to programmatically drive configuration data into its SQL Server repositories.

Control Architect supports Bulk Build/Edit operations using a predefined set of Control Conversion views. To support Control Conversion there are three predefined templates that can be added to a project to define the Property data, the Property bindings, and the Data Source adapter. These templates are referred to as:

- 1. Control Property Template
- 2. Control Binding Template
- 3. Control Conversion Template

When an instance of a template is invoked to view its data, a predefined Template view is used to display its data.

#### **Control Property Template View**

This Template and its view allows you to define the scope of properties (parameters) that represent the source and target of a Control Conversion task. A Property definition requires attributes to be defined such as Name, Description, Size, etc.

Typically the source properties would be developed by pasting Excel Worksheet columns into the Property table or by using a <u>Wizard</u> to extract the Property set from a Honeywell Exception Build (EB) file.

Typically the target properties would be developed by using a <u>Wizard</u> to extract the Property set from an existing C300 Control Module that is being used to represent a typical or template as the target of a Control Conversion task.

#### **Control Binding Template View**

This Template and its view allows you to define Control bindings. A Control Binding defines a relationship or mapping between a source Property and a target Property and contains a binding expression that allows you to define the value conversion if required to convert the value to conform to its target control system platform.

The Control Binding Template requires you to select an existing Control Property Template file for the source and the target of the Control bindings.

#### **Control Conversion Template View**

This Template and its view allows you to define the DataSource (Excel Workbook file) and how data from the DataSource is filtered and/or queried when attached to a particular Template definition within a referenced Control Binding Template.

Typically the DataSource would either be an Excel Workbook file that contains custom engineering data or it could contain data exported from another legacy Control System that is being converted such as a Honeywell Exception Build (EB) file representing Hiway configuration data that was converted to Excel.

The Control Conversion Template view also provides a <u>Control Graph</u> view that allows you to view Control Entity relationships in a graphical view for Honeywell Hiway, Honeywell xPM, and Honeywell AM entity types. This visual Control graph is very helpful in understanding the Control Strategy relationships and the scope when making decisions on how to migrate to another Control System platform.

This view allows you to generate a <u>Data Collection</u> that is subsequently used in a <u>Bulk Build/Edit</u> task.

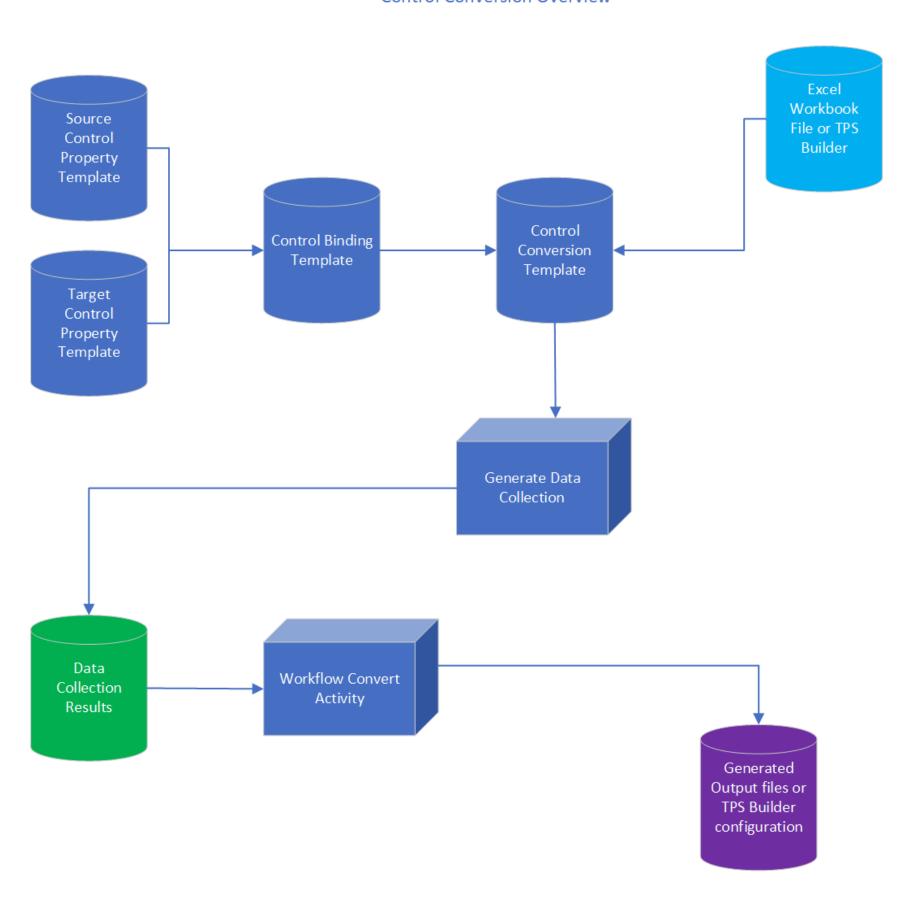
## **Getting Started Using Control Conversion Views**

#### **Overview**

Control Architect provides 3 Control Templates that are used to define a Control Conversion task.

- 1. The Control Property Template defines the Control Properties or Parameters that define a particular Control Template.
- 2. The Control Binding Template defines the relationships or mappings between 2 Control Property templates that represent a set of source properties and target properties.
- 3. The Control Conversion Template defines the output adapter type to use and the Excel Data Source that contains the data to be converted to its target Control System platform.

#### **Control Conversion Overview**



#### **Control Properties**

You can quickly establish your Control Property templates by copying the header values from an Excel Worksheet and pasting them into a Control Property Template using Ctrl+C/Ctrl+V keyboard sequences.

Control Architect provides an Import Wizard that also allows you to quickly connect to Honeywell exported configuration file types including PKS Server CEE exported XML files, PKS Server QuickBuilder exported PNT/HDW text files, LCN Exception Build (EB) text files to establish the target Control Property set for a Template.

Copyright © 2024 by Automation Software Technology, Inc. All Rights Reserved.

The configuration file entity that is used to establish your set of target control properties becomes the basis for your BasedOn Control Template. This Control Template is used when executing Bulk Build/Edit type tasks as the basis for editing or creating new configuration entity instances.

#### **Control Bindings**

A Control Binding Template establishes a relationship or mapping between 2 Control Property Templates that represent the source of the Property or Parameter data and the Property or Parameter data for the target Control System platform.

A Control Binding establishes a relationship or mapping between a particular source Property and a target Property. A default C# script is automatically generated for a Control Binding when the source and target properties have been defined. You can change the results of the Property get accessor in the generated script to provide custom transformation of the source Property value during conversion.

Each Control Binding Template instance must define a *Primary Key* attribute so that the Control Conversion task can uniquely identify the data contained in the Excel Data Source. This will almost always be the entity *NAME* data property.

The Control Binding Template also contains a button target that will automatically generate a list of Control bindings based on exact source and target Property name matching. It will also automatically set the *Primary key* binding for a Control Binding whose source Property name equates to *NAME*.

You can view/edit the Control Binding script by left mouse double-clicking on the column named *Code*. This will expand a details view of the Control Binding item and display the script code in a C# Code Editor View. To collapse the detailed script view double-click on the Code column. Hovering your mouse cursor over a Control Binding items Code column will also show a ToolTip of the script code.

#### **Control Binding Evaluation**

When modifying the default generated Control Binding scripts you should always compile and evaluate the script to validate its script code changes. Select a particular Control Binding or use the key sequence **Ctrl+A** while focused in the Control bindings DataGrid view to select all Control Binding items. Right mouse-click to display a shortcut menu and select the menu item **Compile and Evaluate Property Snippet** to compile and generate a set of properties to be tested in the Evaluate View. After compilation has completed successfully the Evaluation View will be popped into view. You will enter values in the *Source Value* column and verify that the *Target Value* column displays the proper transformed results.

If a code compilation fails, the Output View will show a Flashing LightRed background color to indicate that errors were encountered during compilation. The compilation error usually indicates the source code line number where the error occurred. You can view the full compiled C# script by clicking on the Code View and looking at the referenced source code line number. This Code View is a read-only view of the compiled source code. You will need to open the C# script editor for the Control Binding that contains the errors and modify its script contents.

#### **Control Conversion**

A Control Conversion Template establishes a relationship between a Control Binding Template instance and an Excel File Data Source. It also defines the type of data to be generated by specifying the type of Output Adapter to use. This release of Control Architect only supports PKS XML configuration as the output adapter type.

The basis of all Bulk Build and Bulk Edit tasks is the Control Conversion Template. The execution of a Control Conversion task from within the Control Conversion Template view will generate a Data Collection View that contains the results of processing Control binding evaluations against your Excel Data Source and referenced Control BasedOn Template files.

The saved results of the generated Collection View are used in a Workflow Control Conversion Activity to update existing or create new control configuration entities.

#### **Control Conversion Workflow Activity**

A Workflow activity uses the Data Collection results to generate output based on the type of Output Adapter that was chosen in the Control Conversion Template.

Workflow activity types for Control Conversion include:

- PKS C300 or ACE CEE configuration generation
- PKS Scada configuration generation
- LCN EB file based configuration generation

## **Control Property View**

The Control Property View is where Control Properties (Control Strategy parameters) are defined for use in Control Conversion type tasks.

A Control Strategy uses what is typically referred to as parameters to define its configuration. In Control Architect a control strategy parameter is referred to as a Control Property. Defining a set of control properties allows you to define a control binding that maps the transfer of data between a set of control properties. For example to define a control binding between a value that exists in an Excel Worksheet that has a column name of *PVEUHI* and a C300 Process Regulatory Control Strategy that contains a Regulatory PID function block type named *PIDA*, two control properties would be defined as follows:

- 1. Excel DataSource Control Property = PVEUHI
- 2. C300 Control Strategy Control Property = PIDA.PVEUHI

When defining a Control Binding, the source property would be PVEUHI and the bound to target property would be PIDA.PVEUHI.

#### **Control Property Template**

Control Architect installs a default Control Property template that can be <u>added</u> to a Solution View project.

You can define multiple templates within the same Control Property template. For example you might define a Template that contains properties for all Regulatory Control template types available in the C300. Another Control Property template might define all the property types for Digital template types such as DevCtl, Flag, Digital Input, etc.

#### **Using the Control Property View**

#### Tips and Tricks

- To rename the block identifier for all similar control properties, a menu will become visible after the block identifier portion of a control property has been changed. Click this menu item **Rename all Properties with same [Name].**
- You can copy and paste a range of values from an Excel document into the Control Property view to quickly build a set of properties. Select the range of property names from the Excel Worksheet, press **Ctrl+C**, focus in the Control Property view by clicking anywhere in the Property DataGrid or select the Template in the Template DataGrid, then press **Ctrl+V** to paste the contents of the clipboard. Any leading or trailing white space found in the Excel column name will be stripped when pasting into the Control Property Template to form a valid Control Property name.
- You can copy a range of existing Control Property items in the Property DataGrid using **Ctrl+C** and paste them back into the Property DataGrid using **Ctrl+V** to create copies of the Control Property items.

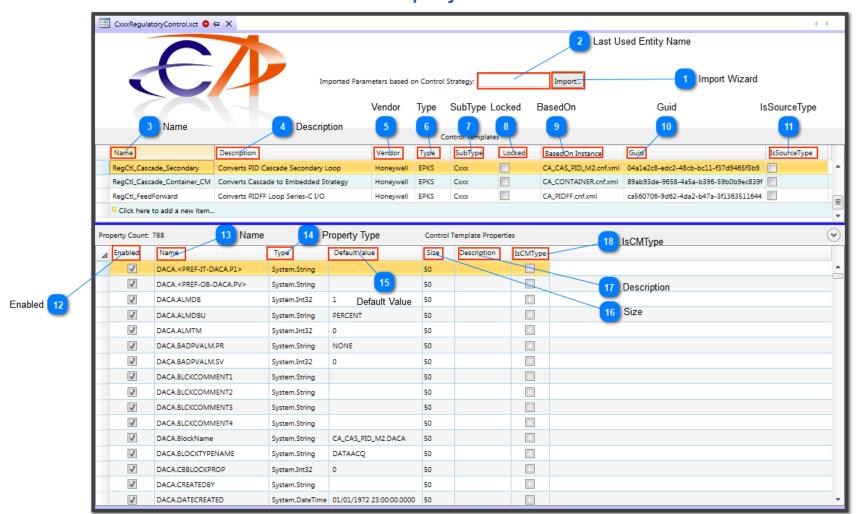
#### **Shortcuts**

- To quickly enable/disable a group of control property items, select multiple items and press the [Space] key to toggle their current state.
- To quickly define a group of control property items as a Control Module type property, select multiple items and press the [Shift] + [Space] key to toggle their current state.

#### Notes:

- Do not use square brackets when forming your Excel column names. Instead of CISRC[1], use CISRC(1) or CISRC1. The Control Conversion Template <a href="DataView">DataView</a> will not display Excel data whose column name contains square brackets.
- Target Control Property names cannot contain any parenthesis characters (). The Import Wizard automatically replaces any array type parameters that include the characters "()" with the bracket style "[]". Square brackets are supported for target property names, but not source property names. The Excel Data Source is read using SQL query syntax where brackets are special characters that cannot be used. However the target Control Strategy data or files are not processed using any SQL technology, but uses XML query technology that does not have a limitation on the use of "[]" brackets for array type parameter or property names. Based on the state of the IsSourceType option, validation of the Property name will show an error image in the NAME cell if invalid usage of array type characters are detected.
- Control Binding relationships are forward only from the source to the target.
- When PKS Cxxx/ACE configuration is chosen as the output target during a Control Conversion task, you need to assure that the Control Property item being referenced as the target has its IsCMType attribute properly chosen. If the Property (Parameter) resides on the Control Module block itself, this attribute must be checked or the Control Conversion task will fail to locate and process the Control Binding.
- 1 If you change an existing Control Property definition that has already been used in a Control Binding, you may need to regenerate the Control Binding snippet code to pick up any changes that were made such as changing the status of *IsCMType*. The properties that make up a Control Property definition are generated to the code snippet as attributes that may require updating by either regenerating the code snippet or manually editing the code snippet source code.

## **Control Property Overview**



Import Wizard

[Import...]

Use the Import Wizard to quickly define sets of Control properties based on existing entities that may reside in an EB file, EPKS ERDB, PKS Xml Configuration file.

2 Last Used Entity Name

The last used entity name is displayed here for reference after an import using the Import Wizard has completed.

3 Name

The name of the Control Property template. This name must be unique for the set of templates within this template file.

Description

Description

Specify a description for this Control Property template.

5 Vendor

Specify a vendor name for this template.

Type

Type

Specify a type name for this template.

7 SubType
SubType

Specify a sub type for this template.

8 Locked

Check this to lock down the template to prevent any further changes.

9

#### **BasedOn**

BasedOn Instance

This reference is automatically set when the Import Wizard is used to generate a set of Control properties for the template.

10

#### Guid

Guid

This is a unique identifier for this template that is auto-generated.

11

#### **IsSourceType**

IsSourceType

Choose this option if the Control properties defined for this Template represent your Excel Data Source. This option is used to validate Property name constraints such as the use of characters when specifying array types as in CISRC(1). Excel column data cannot have the bracket chars "[]" in the name. An Excel name such as CISRC[1] will fail on it data retrieval as the bracket characters have special usage when defining queries against the Excel Data Source. Control Architect expects that target Control properties that require array access be named using the square brackets as in CISRC[1]. But this bracket style only applies to target type Control properties.

Leave this option unchecked if the properties represent a target Control System platform.

12

#### **Enabled**

Enabled

Check to allow this property to become visible in drop-down lists when this template is referenced in a Control Binding template.

13

#### Name

Name

The name of the Control Property. This name must be unique within the current Control Property template.

14

#### **Property Type**

Type

Choose the Property Type from the drop-down list of available types.

15

#### **Default Value**

DefaultValue

Specify a default value to use when participating in a Control Binding relationship.

16

#### Size

Size

Specify a size value for String type properties that defines the maximum length of a string value. This size constraint only applies to System. String type properties.

17

#### **Description**

Description

Specify a description for this Control Property. When importing from an EPKS ERDB database, this description property is automatically imported.

18

## **IsCMType**

IsCMType

Check this if this target Control Property resides on the Control Module block itself and not on an embedded function block. When developing properties for Honeywell PKS configuration this attribute only applies to the target PKS Control Module Property. There is no need to set it on the Excel DataSource Control Property definition.

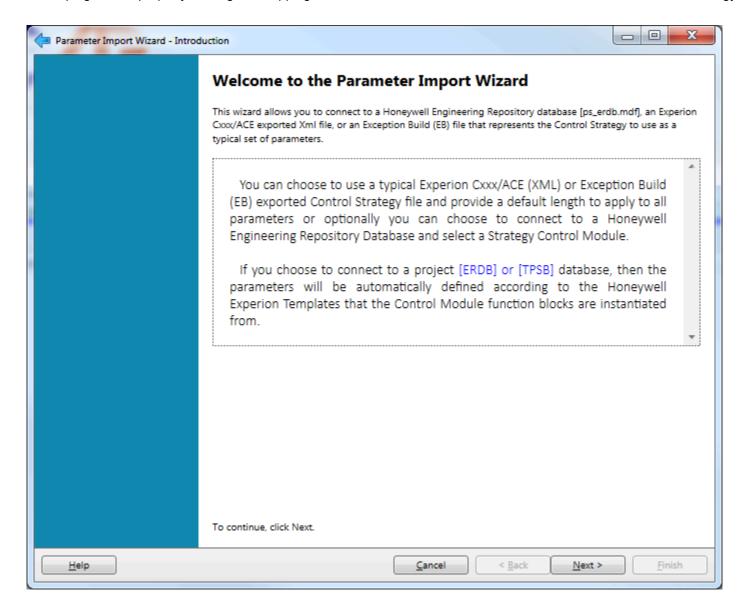
## **Importing Parameter References**

## **Parameter Import Wizard**

This wizard guides you through importing parameter references from multiple types of sources including:

- Honeywell Exception Build EB files
- Honeywell PKS XML configuration file
- Honeywell Experion Server

Using one of these parameter reference sources will allow you to quickly define a set of control properties that can be subsequently used when developing control property bindings or mappings between an Excel DataSource and Cxxx/ACE resident control strategy data.



#### **Using the Wizard**

#### Tips and Tricks

- To quickly change the parameter type for multiple parameter items, multiple select items in the Parameter List and while keeping pressed the [Ctrl] key, select a new parameter type from the dropdown list.
- To quickly change the parameter size for string type parameter items, multiple select items in the Parameter List, change the size value and hold the [Ctrl] key while pressing the [Enter] key.
- Use the filter box to enter text to filter the list of parameter items. Text that is entered will filter the parameter items if the parameter name contains any portion of the filter text.

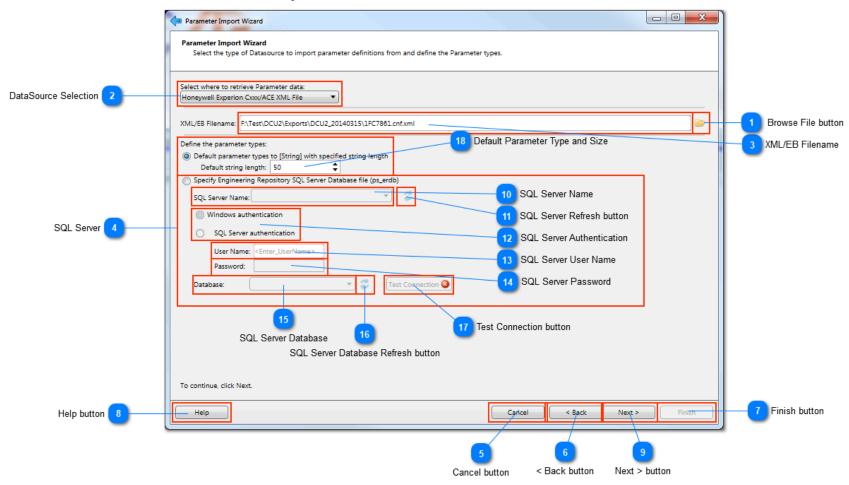
#### **Shortcuts**

• To quickly enable/disable a group of parameter items to import, select multiple items and press the [Space] key to toggle their current state.

#### Notes:

• To select all parameter items using the key combination **Ctrl+A** select a parameter item first, which may require 2 clicks to take focus away from other column controls.

#### **Import Wizard DataSource selection**



## Browse File button



Click this button to browse to a file that contains the DataSource file that will be used to import parameter data from.

## DataSource Selection



Choose from the drop-down list the type of DataSource to use to import parameter data from. The options are *Honeywell Experion Cxxx/ACE XML File*, *Honeywell Exception Build (EB) File*, *Honeywell Experion SCADA PNT file*, and *Win7 TPS Builder SQL Server*.

## XML/EB Filename

F:\Test\DCU2\Exports\DCU2\_20140315\1FC7861.cnf.xml

This is the current file path where the selected DataSource file resides.

## SQL Server



Specify the Server connection data if importing parameter data from an SQL Server ERDB or TPS Builder Database.

# Cancel button



Click this button to cancel the current Import Wizard task and close the window.

# 6 < Back button

Click this button to navigate to the previous Wizard page.

Copyright © 2024 by Automation Software Technology, Inc. All Rights Reserved.



Click this button to exit the completed Wizard task.

8 Help button

Click this button to view Help associated with this Wizard page.

9 Next > button

Click this button to navigate to the next Wizard page.

SQL Server Name

SQL Server Name:

Select a SQL Server instance name from the drop-list. This only applies if the ERDB option is chosen.

SQL Server Refresh button

Click this button to refresh the network search for available SQL Server instances.

SQL Server Authentication

Windows authentication

SQL Server authentication

Choose the type of authentication required to login to the selected SQL Server instance.

SQL Server User Name

User Name: <Enter\_UserName>

Specify the User Name for logging into the selected SQL Server instance.

SQL Server Password

Password:

Specify the Password for logging into the selected SQL Server instance.

SQL Server Database

Database:

Specify the SQL Server Database that identifies the Honeywell ERDB for the selected SQL Server instance.

SQL Server Database Refresh button

Click this button to refresh the list of available databases for the selected SQL Server instance.

Test Connection button

Test Connection 3

Click this button to test the connection to the selected SQL Server instance using the provided login credentials.

Default Parameter Type and Size

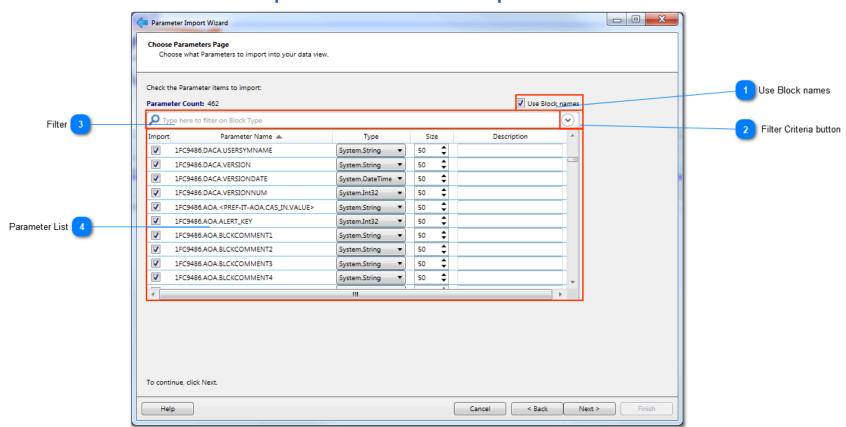
Define the parameter types:

Default parameter types to [String] with specified string length

Default string length: 50 \$

Choose this option to default all parameter types to a String type with the specified length in characters.

## **Import Wizard choose XML parameters**



# Use Block names

Use Block names

Choose this option to qualify the parameter items using the block identifier to form a composite BlockName.ParameterName.

# Filter Criteria button



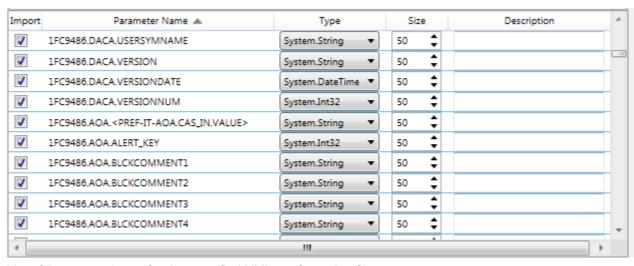
Click this drop-down button to choose filter options.

Filter



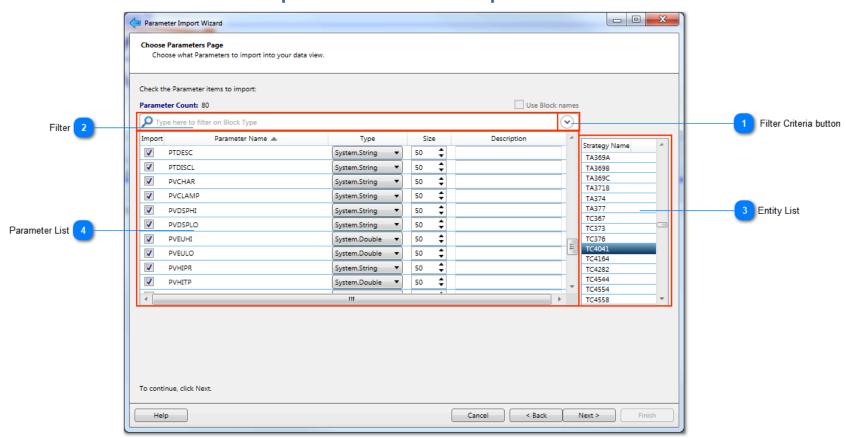
Enter a filter expression to filter on the Parameter items.

#### Parameter List



List of Parameter items for the specified XML configuration file.

## **Import Wizard choose EB parameters**



## Filter Criteria button



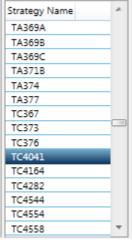
Click this drop-down button to choose filter options.

Filter

Type here to filter on Block Type

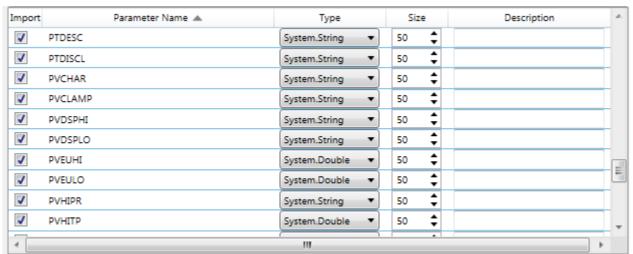
Enter a filter expression to filter on the Parameter items.

## Entity List



List of entities that were parsed from the specified EB file.

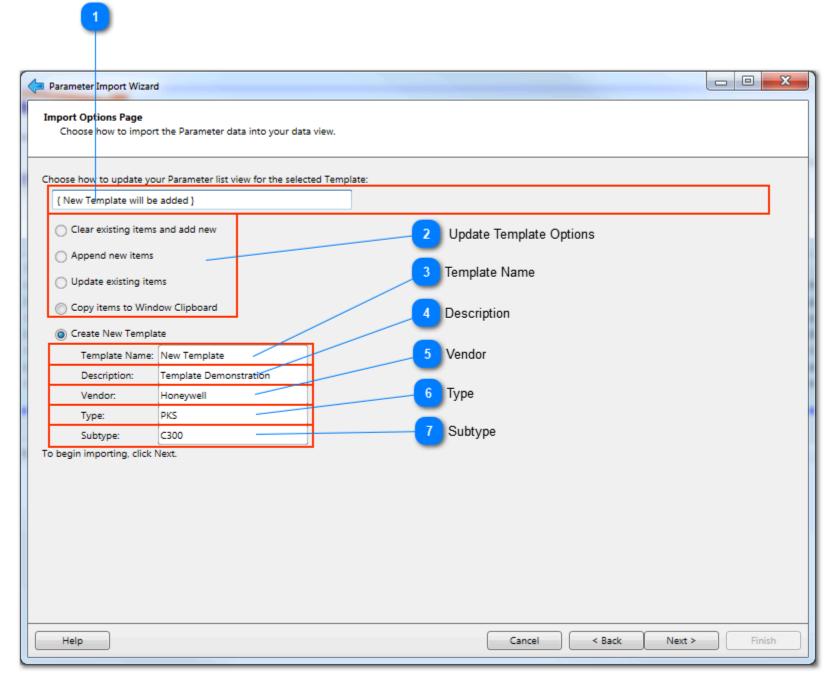
#### Parameter List



List of Parameter items for the selected EB Entity.

## **Import Wizard specify parameters Template**



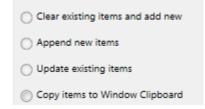


## Import to Template Status

{ New Template will be added }

Displays a description of the activity that will occur during import of the parameters to a Template.

## Update Template Options



Choose how to update the selected Template in the Control Property view. If no Template was selected when this Import Wizard was activated, then the only applicable option is *Create new Template*. The default option for overwriting an existing Template is to *Append new items*.

The option "Copy items to Window Clipboard" is used internally by Control Architect functions and its Clipboard content has no real value or purpose to the user. When invoking this Wizard view from the "Create Control Conversion Project" Workflow view this option is automatically chosen and cannot be changed. The Control Property set defined here is copied to the Window Clipboard and its contents is read to develop the list of target Control properties.

#### Template Name

Template Name: New Template

Specify the name of the new Control Property Template.

Description

Description: Template Demonstration

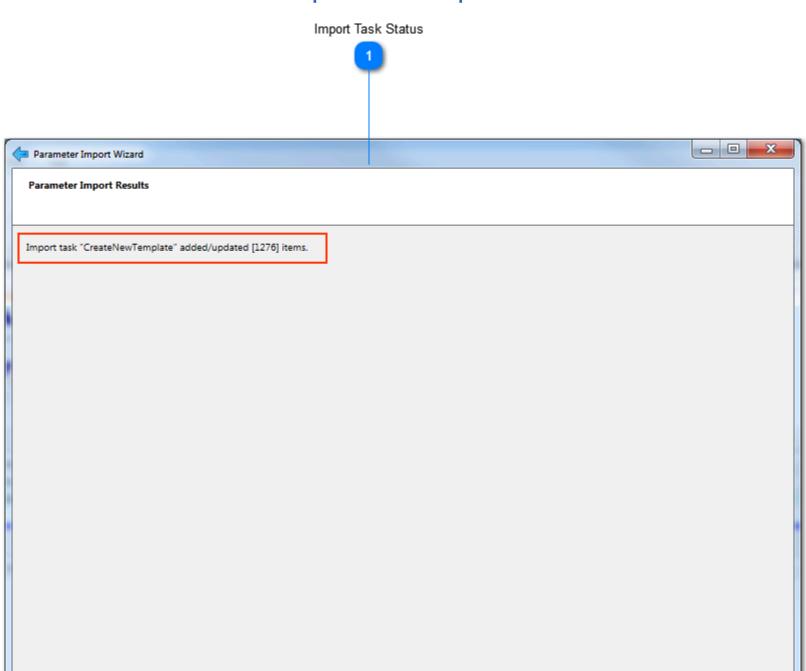
Specify the description of the new Control Property Template.







# **Import Wizard completion**



< Back

Cancel

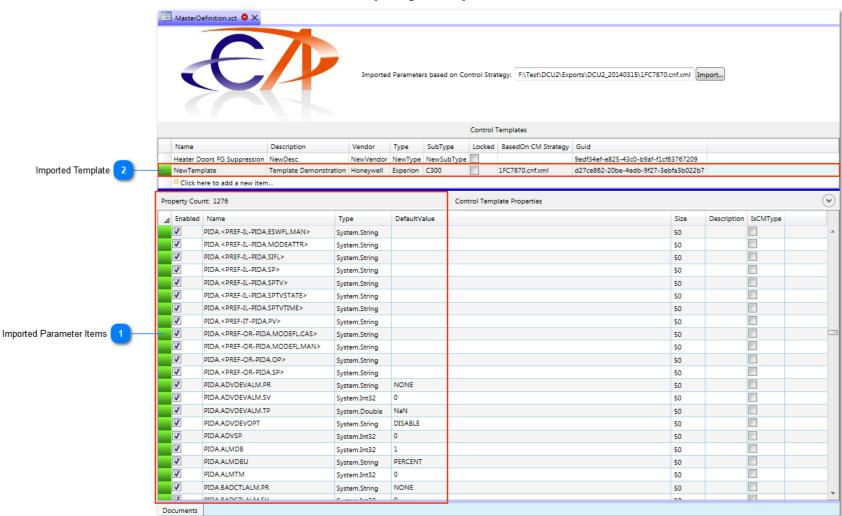
Next >



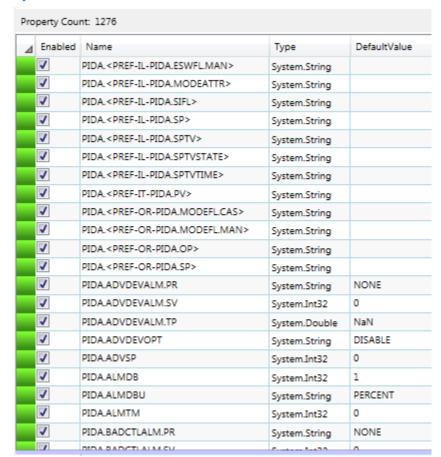
 $Import\ task\ "CreateNewTemplate"\ added/updated\ [1276]\ items.$ 

Displays the results of the Parameter import.

#### **Control Property Template results**



## Imported Parameter Items



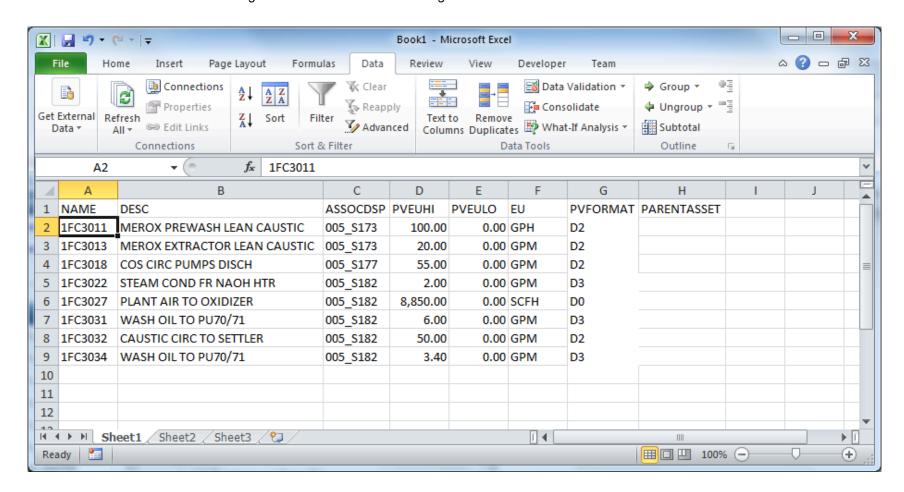
The green row header status shows the new Parameter items that were added during the Import Wizard session.



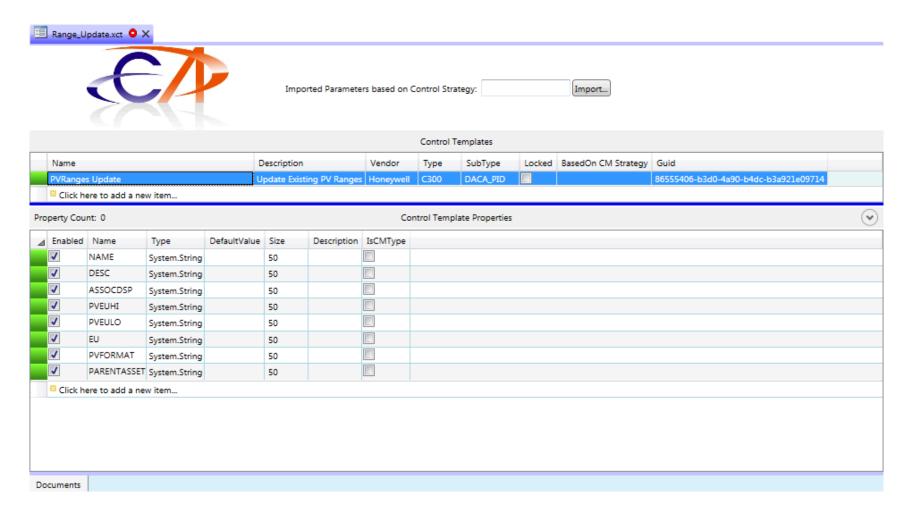
The green row header status show the new Template that was added during the Import Wizard session

## **Example Excel Workbook Data Source**

Here is an example of a simple Microsoft Excel Workbook being used to update existing Control Module strategies. This Excel Workbook is used as the DataSource to bind to target Control Module based strategies.



Here is an example where the Excel header row columns above where copied and pasted into a new Control Property Template to establish the source Control Property Template to use when defining the Control Binding Template.

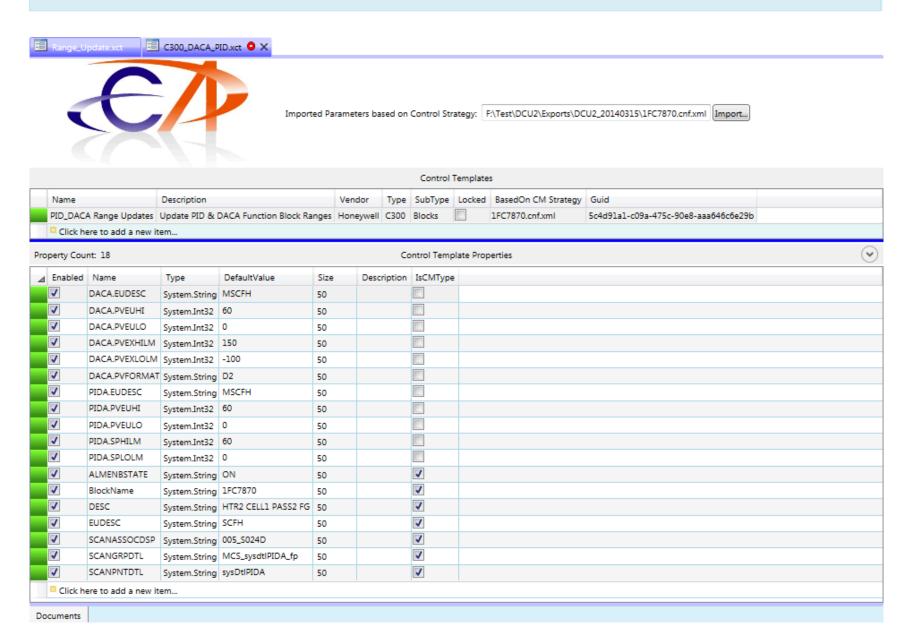


Here is an example where another Control Property Template was added to a Solution project. The Import Wizard was used to import parameters from an existing Control Module that contained a PID and DataAcq function block to establish the target Control properties that will be referenced in the Control Binding Template. Notice that when importing from the Import Wizard where the properties are defined from an existing PKS Cxxx/ACE type Control Module, the IsCMType attribute for the new Control Property item is automatically checked based on whether the Property existing at the Control Module block itself.

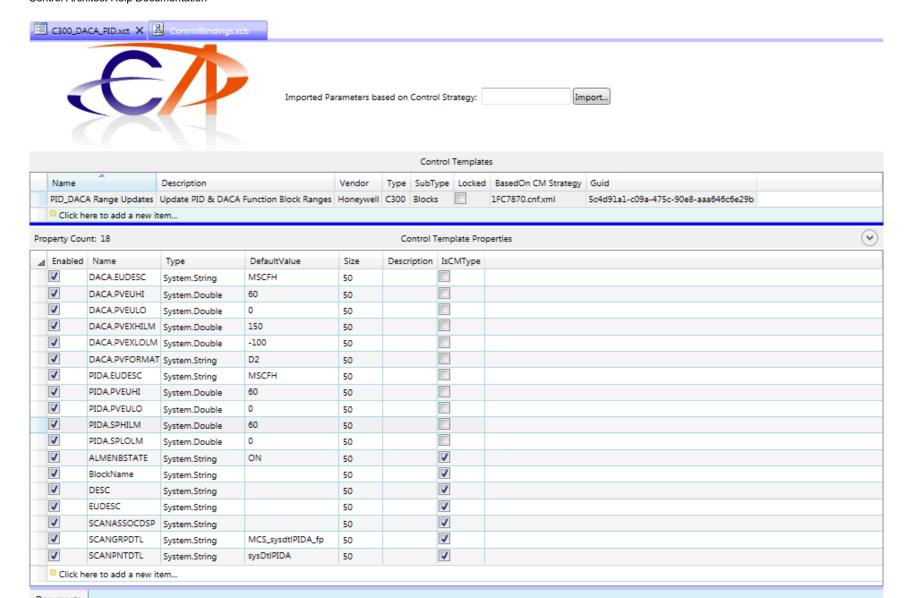


The Import Wizard will attempt to identify the Property type based on its current value. Values such as 100 or 0 would be interpreted to be an *Integer* type where as a value such as 100.1 would be interpreted as a *Double* type. The properties below for

the PV type of parameters were imported as *Int32* types and need to be changed to *Double* types manually after they have been imported. We could have optionally changed the type within the Import Wizard prior to importing.



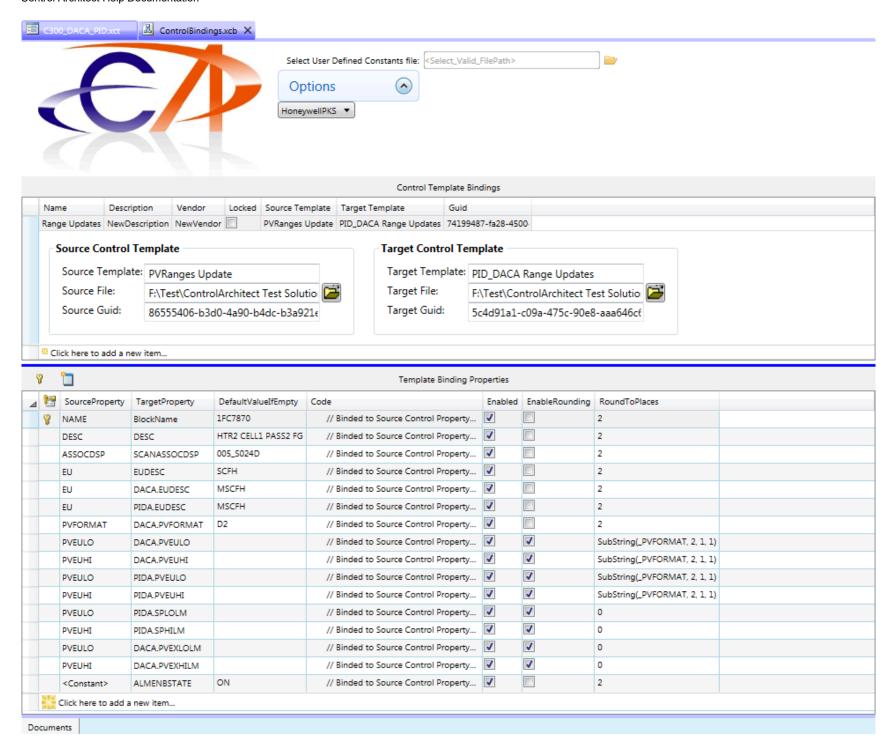
This is the same Control Property view as above, where we modified specific Property types to reflect double types and removed some default values.



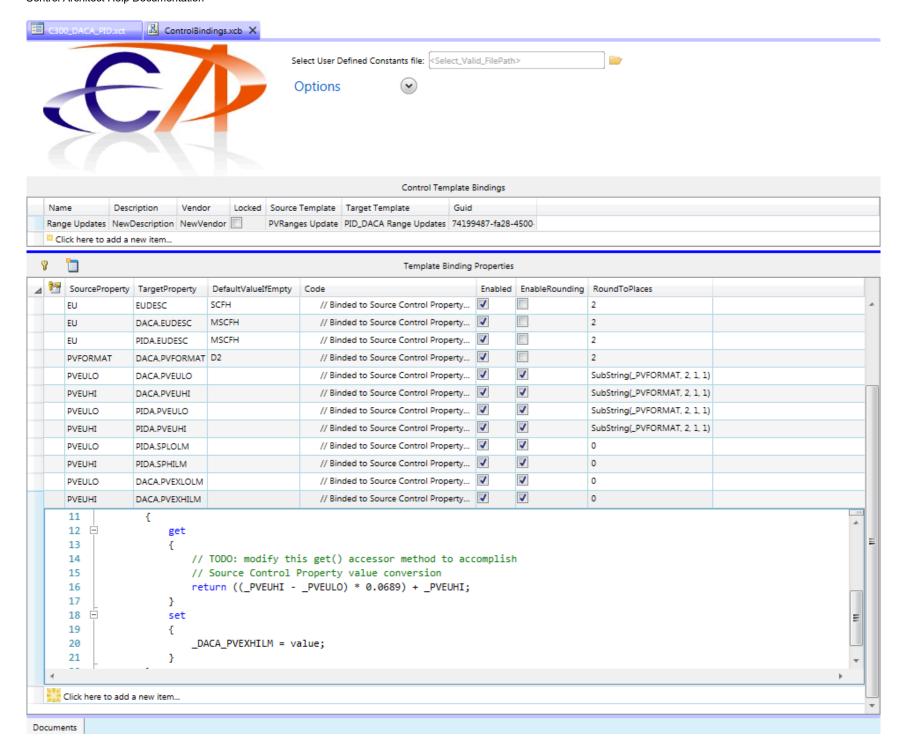
Here is an example where a Control Binding Template was added to a Solution project that references the two Control Property Template files shown above as the source and target Control properties. Double-click on the Template item to display the items detail data.



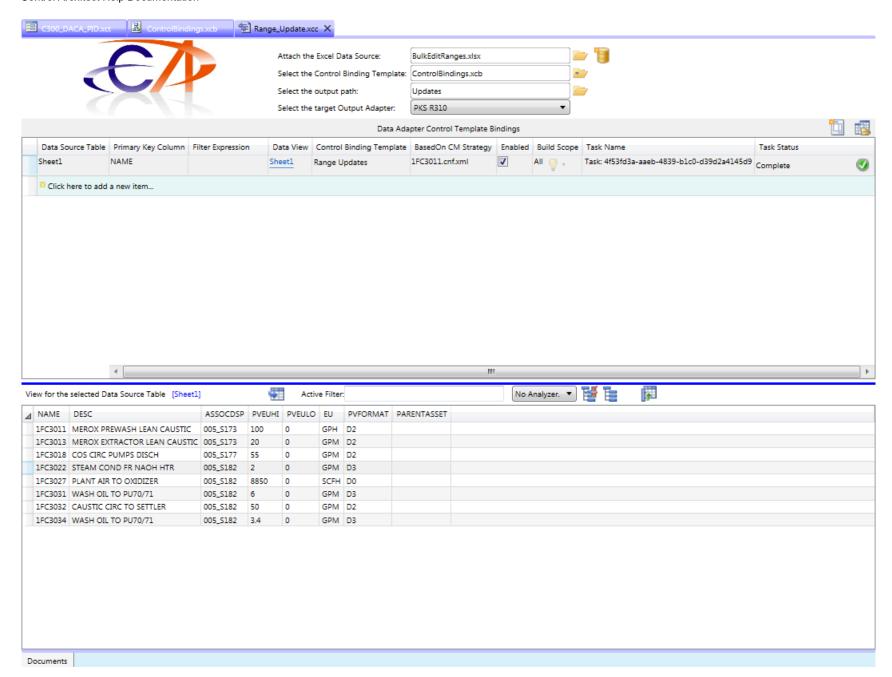
You must define a specific Control Binding in the set of Control bindings as a Primary Key. This Primary Key defines which source Control Property uniquely identifies the data found in the Excel DataSource. In most cases this will normally be the entity name of the Strategy that is being edited.



Here we have modified the default generated Control Binding expression to transform the value for the target Property PVEXHILM based on the values of PVEULO/PVEUHI. The default generated expressions simply pass through the value found in the DataSource to the target Property. If a value requires transformation to another value based on the values of other referenced source or target Control properties, you will have to modify this default conversion expression.



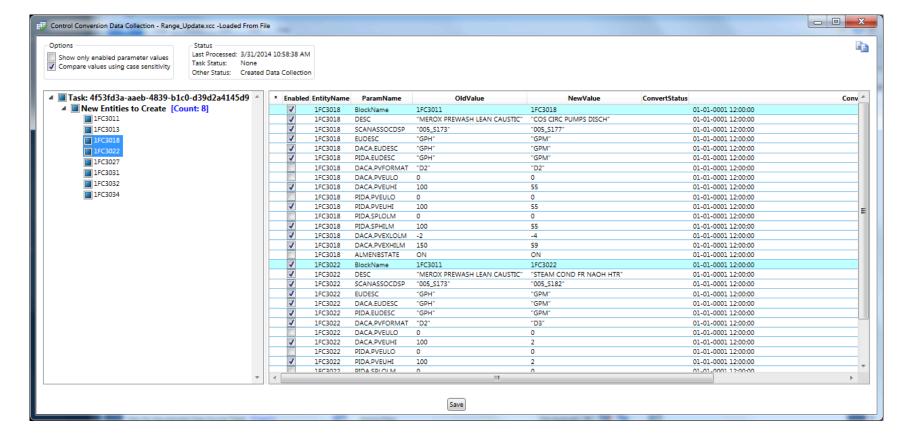
Here we added a new Control Conversion Template to the Solution project that allows us to bind our Control Binding Template above to an Excel DataSource.



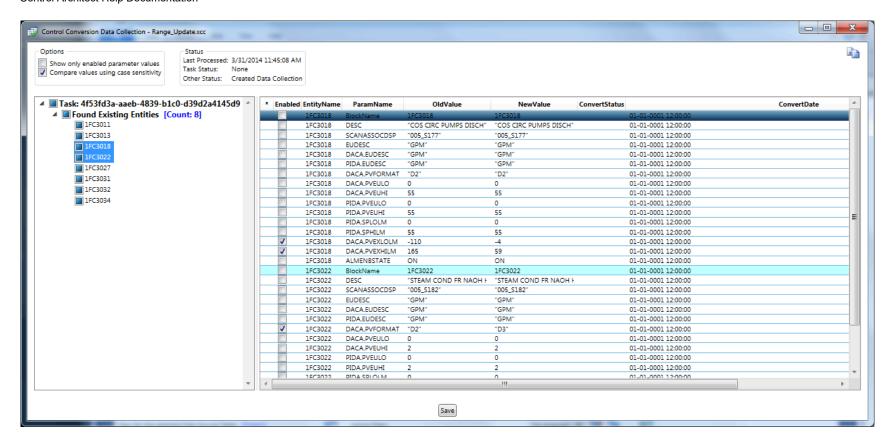
Here is an example of executing a Data Collection task using the Control Conversion Template defined above.



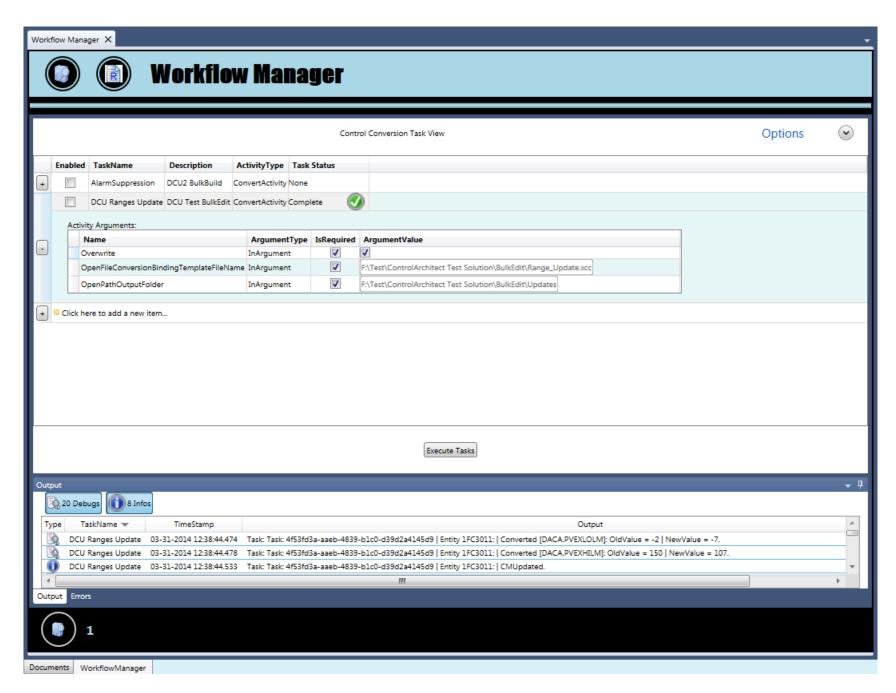
The Data Collection indicates that the referenced DataSource entities will be created as new because the Control Module strategies do not exist in the specified *Output* folder.



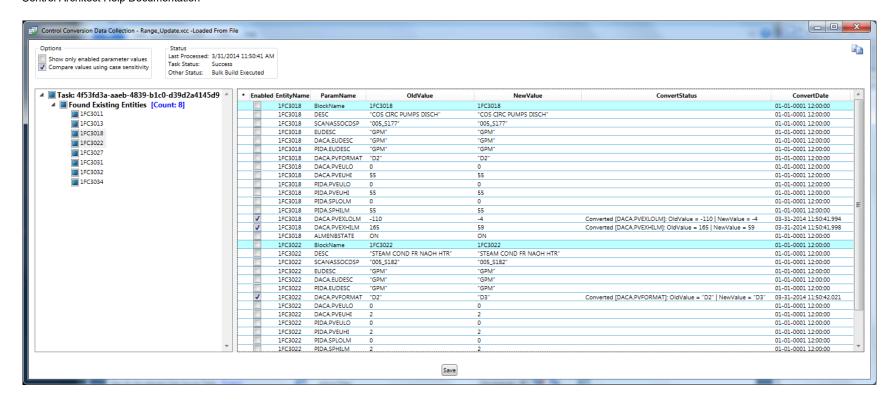
Here is another example of executing a Data Collection task using the Control Conversion Template defined above, but this time we placed the existing Control Module XML files in the *Output* folder in order to update the existing entities.



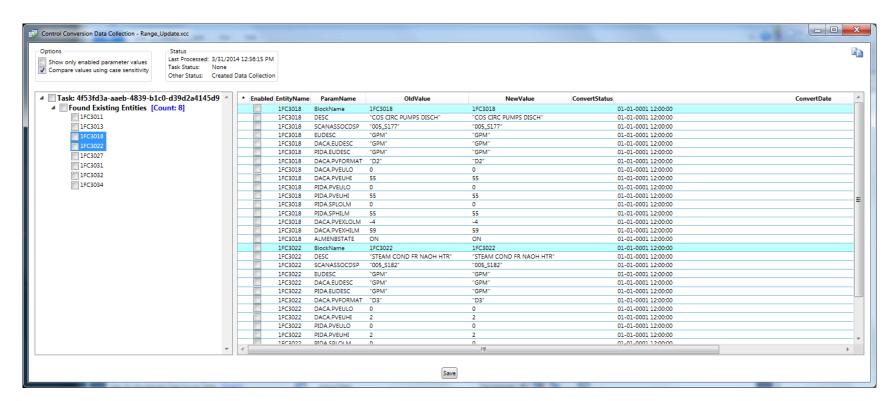
This is an example of defining a Workflow task using the Control Conversion Template above and its saved Data Collection results. When this Workflow Control Conversion task is executed, the target of the conversion task is created or updated to reflect the changes that were defined in the Data Collection.



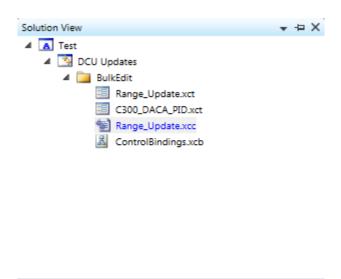
Here is the same Data Collection above that has been executed in a Workflow task. The Parameter items that have been updated during execution of the Control Conversion task are shown here with the results of the edit operation.



If we were to again execute a Data Collection task on the Strategy items that were just updated, we would validate that the target strategies are now updated to the values defined in the Excel DataSource as no Parameter item values are indicated as changed.



Here is the Solution workspace that was used to contain the Control template developed above.



# **Control Binding view**

## **Overview**

Control Binding templates allow you to define a set of bindings that define how data located in a source Control Property template is mapped to data located in a target Control Property template.

Bindings can be simple as mapping a source Property name (parameter) to a target Property name (parameter) without any data conversion. Complex conversion expressions can be developed to support the type of data conversion required when reading the source Property name value and converting this to a value required by the target of the conversion.

## **Control Binding Template**

Control Architect installs a default Control Binding Template that can be <u>added</u> to a Solution View project. Using an added instance of this template will allow you to bind the Template to existing instances of <u>Control Property</u> templates that define the properties for the Data Source and the properties for the target Control System strategies.

You can define multiple template bindings within the same Control Binding Template. For example you might define a Template binding for handling simple PID type Control strategies and another Template binding for handling simple PID Cascade type Control strategies.

# **Working with Control Property Bindings**

Control Property bindings define the mapping or relationship between a Property from a Data Source such as Excel and a Property or parameter in the target Control system Template type. Control system Template types contain properties (parameters) that define its class.

# **Primary Key Attribute**

You must specify a single Control Binding with the Primary Key attribute. This is the Control Binding that defines the unique Entity name for the Control Entity that is being updated/created. For the Honeywell platforms this would be the Target Property named "NAME" for LCN types and for PKS CEE Control Modules this would be the "BlockName" Target Property. Click the button target to add the Primary Key attribute to the selected Control Binding.

## **Using the Rounding Options**

Target properties that are typed as *float* or *double* can optionally be rounded using either a constant value or specifying a rounding function based on a source Property value.

The selected item below indicates that the rounding expression uses the *SubString(...)* function that is recognized by Control Architect. This function uses the following parameters to extract the decimal rounding value from a source property value retrieved from the source Property name *\_PVFORMAT*.

SubString(sourcePropertyName, startingPosition, count, defaultValue)

sourcePropertyName: Specify the source Property name that will contain the expected format value. Honeywell Control types use the parameter *PVFORMAT* to specify an enumeration value in the enumeration set of { D0, D1, D2, D3 }. You must prefix the Property name with the '\_' character to indicate it is referencing a source Property.

startingPosition: Specify the one based character starting position where you expect to parse the decimal value from the enumeration value string. If the *PVFORMAT* value contained the enumeration value "D1", the decimal value of "1" begins in character position 2.

count: Specify the number of characters to parse that defines the decimal value in the enumeration value string. If the *PVFORMAT* value contained the enumeration value "D1", we would only need to parse one character to retrieve the full decimal value that defines the rounding expression.

defaultValue: Specify the default rounding expression value to use if the evaluation of the *SubString(...)* function fails to parse a valid decimal value or if the source Property value is an empty string.

Copy this example: SubString (\_PVFORMAT, 2, 1, 1) to use as a rounding expression. The Property *PVFORMAT* must exist as a Source Property.

PVEULO	DACA.PVEXLOLM	-2.9	// Binded to Source Control Property	V	<b>V</b>	SubString(_PVFORMAT, 2, 1, 1)
PVEUHI	DACA.PVEXHILM	102.9	// Binded to Source Control Property	1	V	SubString(_PVFORMAT, 2, 1, 1)
<constant></constant>	DACA.PVSRCOPT	ONLYAUTO	// Binded to Source Control Property	1		2
<constant></constant>	DACA.PVSOURCE	AUTO	// Binded to Source Control Property	1		2
<constant></constant>	DACA.P1FILTTIME	0	// Binded to Source Control Property	1	7	2
PVEUHI	AIChannelA.PVEXEUHI	102.9	// Binded to Source Control Property	1	7	SubString(_PVFORMAT, 2, 1, 1)
PVEULO	AIChannelA.PVEXEULO	-2.9	// Binded to Source Control Property	J	<b>V</b>	SubString(_PVFORMAT, 2, 1, 1)
PVCLAMP	DACA.P1CLAMPOPT	DISABLE	// Binded to Source Control Property	1		2

You can also specify a constant value such as "2" for the rounding expression, which would round the real value to 2 places to the right of the decimal point.



Rounding can be disabled or enabled by checking the Check Box in the column labeled **EnableRounding**.

#### **Common Errors**

# **Control Binding Expressions**

1. When a Control Binding Property Snippet has been generated, changes made to the target Control Property in the Control Property
Template can result in the Control Binding not being evaluated properly. For example, if the defined target Control Property is an Integer type
and already used in a Control Binding, then it's script has already been generated based on an Integer type. Subsequently changing this
Control Property type to a Double will not result in the Control Binding script being updated to reflect a Double type resulting in truncating
floating values to an Integer type when executed in a Control Conversion task. When a Control Binding Template view is opened, it will check
the Target Control property types referenced in existing Control Bindings and show an error image if the current target Control Property type
does not match the Control Binding scripted Property type.

Copyright © 2024 by Automation Software Technology, Inc. All Rights Reserved.

- 2. If the target Property attribute *IsCMType* is changed after a Property Snippet has been generated for a Control Binding, you will have to manually change the Property Binding attribute that exists in the Property Snippet expression. The IsCMType binding attribute [IsCMPropertyAttribute(true)] has a logical parameter true or false that indicates if the target Property resides in the Control Module container itself and not within an embedded Function Block.
- 3. When manually adding a Property such as *AssignedTo* to a PKS Property set, you must be sure to define it as a CM (Control Module) type property.

## **Tips & Tricks**

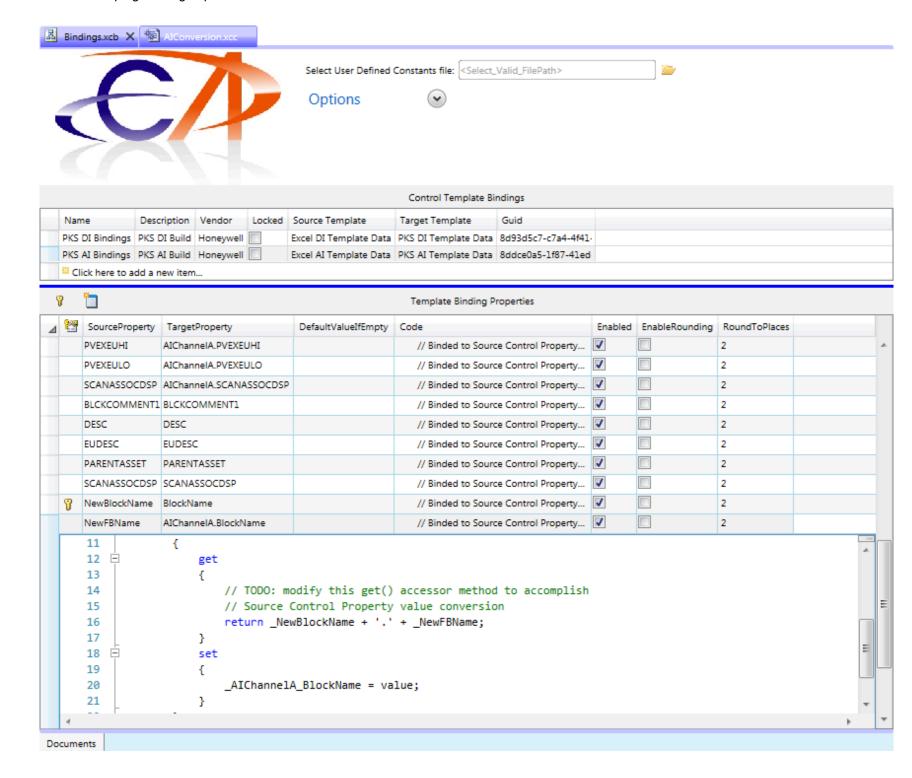
- Use the **F9** Function Key on selected Control Property bindings to set its *DefaultValueIfEmpty* value to the value stored in the Target Control Property reference.
- If you want to specify the Control Execution assignment for a generated Control Module, you may need to manually add the Property name *AssignedTo* to the PKS Property set as a CM String type with a default length of 50 characters. This will allow you to assign the CEE node when generating new Control modules.

### **Function Block Name Change**

You can use the following expression as an example of how to change the name of a Block such as an I/O Channel Block:

```
NewBlockName + '.' + NewFBName
```

Where NewBlockName is the Control Module name Property in the Excel DataSource and NewFBName is the name of the I/O Channel Block Property in the Excel DataSource. The leading '\_' underscore character allows you to reference the source properties by their Property name when developing binding expressions.



# **Control Binding expressions**

# **Building expressions to convert parameter values**

In order to perform a Control conversion task, you must define the conversions or transformation of Parameter values from a source to a target. Where you are converting or importing data from an Instrument database or migrating from one control platform to another there will always be certain Parameter values that will require converting to fit the data to the target Control System platform.

- For example a typical PID algorithm that resides in a Honeywell Basic Controller may use a Control Equation 'B' that when migrating to an HPM or Experion control system will require that Control Equation 'C' is selected to convert to the same control behavior. The Control Binding that is defined for this Parameter item will require that an expression is developed in C# code to support its value conversion.
- When discussing Control binding a Property is the abstract of an algorithm or Function Block Parameter. Parameters defined in a Function Block are technically the properties for the block. The use of the word Parameter is a carry over from legacy Control System configuration.
- A default Property Binding expression that is generated will simply take the value from the source of the Control Binding and provide it unchanged to the target specified in the Control Binding as shown below:

```
// Binded to Source Control Property Value [PVEUHI]
public System.Double Source_AIChannelA_PVEXEUHI;

// Converted Target Control Property
private System.Double _AIChannelA_PVEXEUHI;
[PropertyBinderAttribute("PVEUHI", "AIChannelA.PVEXEUHI")]
[PropertyRoundingAttribute("SubString(_PVFORMAT, 2, 1, 1)", true)]
[PropertyDefaultValueAttribute("102.9")]
public System.Double AIChannelA_PVEXEUHI
{
    get
    {
        // TODO: modify this get() accessor method to accomplish
        // Source Control Property value conversion
        return Source_AIChannelA_PVEXEUHI;
    }
    set
    {
        _AIChannelA_PVEXEUHI = value;
    }
}
```

• Below is a code expression that provides modification to the value of the source Property prior to assigning it to the target Property of the Control Binding. This code expression is using the values of the source properties or parameters PVEUHI and PVEULO to calculate a default value to apply to the target Property PVEXLOLM.



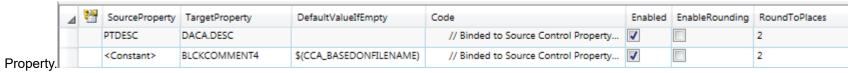
Note that the only valid place to modify a code expression for a Control Binding is in the **get** accessor of the *Converted Target Control Property* as indicated by the //TODO: comment.

```
// Binded to Source Control Property Value [PVEUHI]
public System.Double Source_AIChannelA_PVEXEUHI;
// Converted Target Control Property
private System.Double _AIChannelA_PVEXEUHI;
[PropertyBinderAttribute("PVEUHI", "AIChannelA.PVEXEUHI")]
[PropertyRoundingAttribute("SubString( PVFORMAT, 2, 1, 1)", true)]
[PropertyDefaultValueAttribute("102.9")]
public System.Double AIChannelA PVEXEUHI
  get
    // TODO: modify this get() accessor method to accomplish
    // Source Control Property value conversion
    // Notes: We use a string property to read the enumeration value of PVRNGOP.
    // Note the use of variable names prefixed with the '_' character. You can use a SourceProperty
     // reference by prefixing its name with the '_' character.
    // Note that we use ToUpper() method to compare both values in upper case to eliminate having
    // to worry about case sensitivity.
    return PVRNGOP.ToUpper() == "NONE"?
            ((_PVEUHI - _PVEULO) * 0.0689) + _PVEUHI
```

```
: ((_PVEUHI - _PVEULO) * 0.029) + _PVEUHI;
}
set
{
    _AIChannelA_PVEXEUHI = value;
}
```

#### **Constants**

• Constants can be specified as the source of Control bindings by simply selecting the **<Constant>** as the source Property. The value specified as its default will become the Constant value that is written to the Control Binding target Property during Control conversion execution. You can reference a <u>custom variable</u> as the source of the default value for a **<Constant>** type source



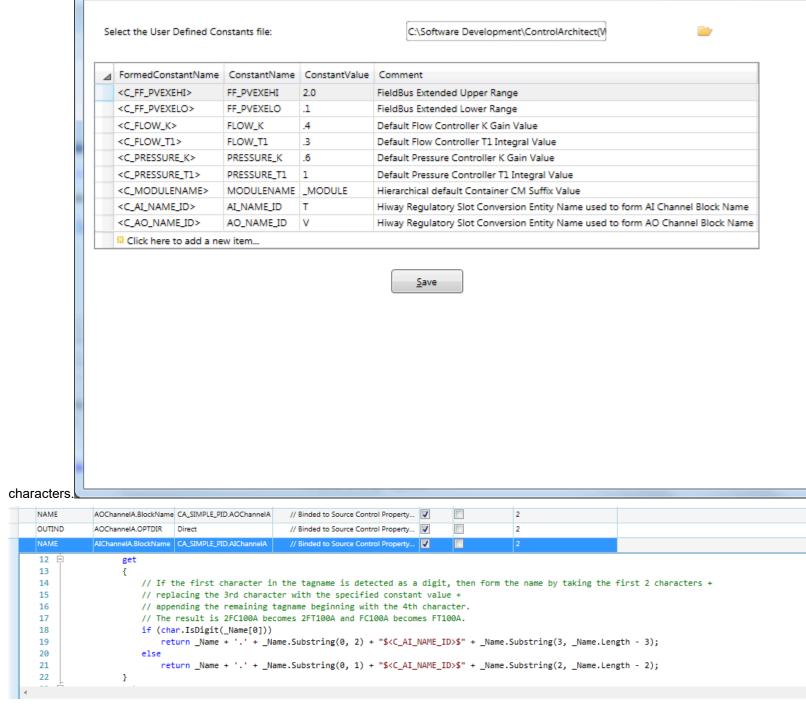
\_ D X

• User defined constants can be referenced in Control Binding code expressions such as:

```
return - (Source DACA PVEUHI - Source DACA PVEULO) * $<C FF PVEXEHI>$;
```

Constants used within script code must be enclosed within '\$'

User Constants Editor

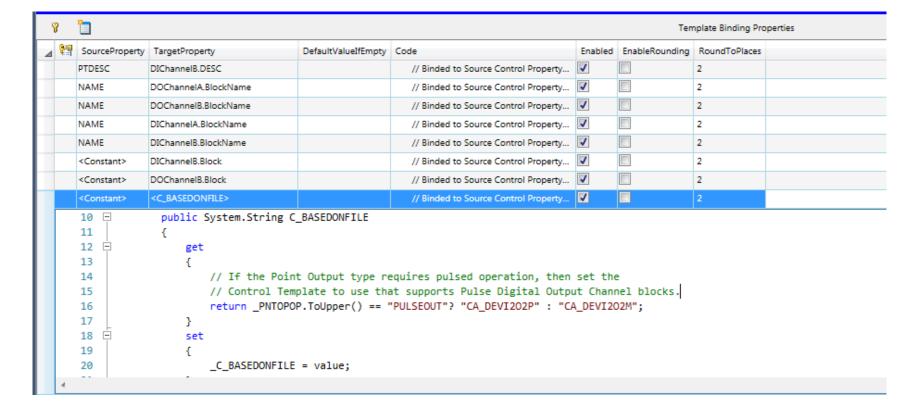


- Users can develop a set of constants that are then referenced by the Control Binding templates. Using the <u>Constants Editor</u> in Control Architect *constants* can be defined that can then be referenced as source properties of Control bindings.
- Constants when defined will be named using <C\_UserName> where UserName specifies the unique identifier for this Constant. Typically a set of constants will be developed that defines the rules for a particular project or client site. Examples of common constants that may be defined include:
  - FFPVExtendedRangeUpperLimit Foundation FieldBus type Process Measure Input Extended Upper Limit
  - FFPVExtendedRangeLowerLimit Foundation FieldBus type Process Measure Input Extended Lower Limit
  - AnalogPVExtendedRangeUpperLimit Standard 4-20 ma device type Process Measure Input Extended Upper Limit
  - AnalogPVExtendedRangeLowerLimit Standard 4-20 ma device type Process Measure Input Extended Lower Limit

- FlowPVLowCutOff standard flow range percent Low Cutoff
- K\_FlowControllerGain standard default Control Gain (K) value for flow service
- T1\_FlowControllerIntegral standard default Control Integral (T1) value for flow service
- K\_PressureControllerGain standard default Control Gain (K) value for pressure service
- T1 PressureControllerIntegral standard default Control Integral (T1) value for pressure service

## **Special Constants**

• Use **<Constant>** as the Source Property bound to **<C\_BASEDONFILE>** as the Target Property in a Control Binding to overwrite the BasedOn Template filename that was used to define the set of Target Control properties. You would specify the pure Control Module name as the value without the file path and filename extension. The Control Module name specified as the value must reside in the folder path that is specified as your PKS Control Templates folder, which can be accessed from the Ribbon Toolbar File menu item.



## **Searching and Replacing Parameter Values**

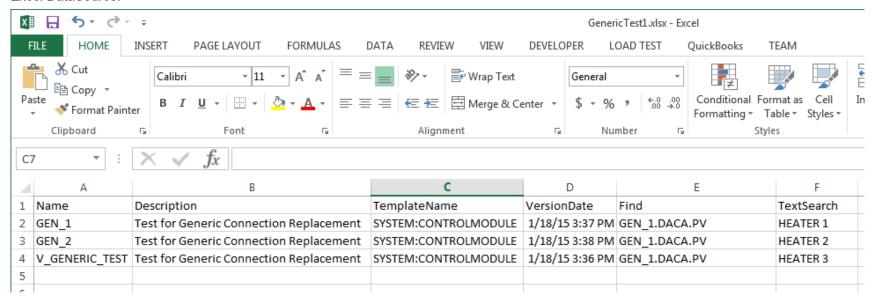
You can search and replace Parameter values by specifying the value of *BLOCK*.</PARAMETER> for the Target Control Property. You define this special Target Property in your Control Property Template as in the example below, which then allows you to select it for the Target Control Property in a Control Binding:

When searching and replacing Parameter values you will define a column in your Excel DataSource such as "TextSearch" that is used to specify the Parameter value to search for. The DefaultValueIfEmpty attribute for your Control Binding is used to specify the replacement value. When this type of Control Binding usage is detected during generation of a Data Collection task all Parameter values for all Control Modules and their embedded blocks will be searched for the value specified in the Excel Data Source.

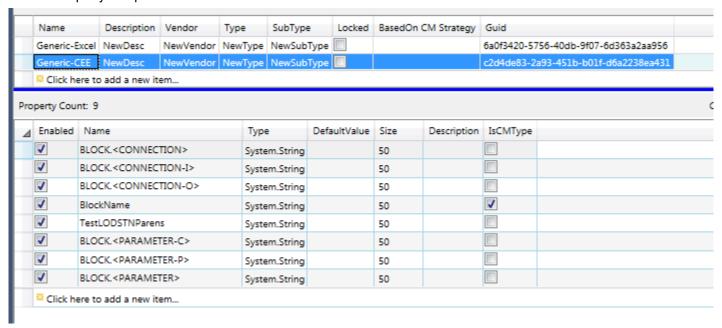
You can specify optional attributes in the Parameter name format:

- BLOCK.<PARAMETER-I> indicates to search for Parameter values while ignoring alpha case. This is the same as formatting the Property as BLOCK.<PARAMETER>.
- BLOCK.<PARAMETER-C> indicates to search for Parameter values using alpha case sensitivity.
- BLOCK.<PARAMETER-P> indicates to search for partial use of the specified Parameter value while ignoring alpha case. This would often be used to edit values specified in function descriptors, keywords, etc.

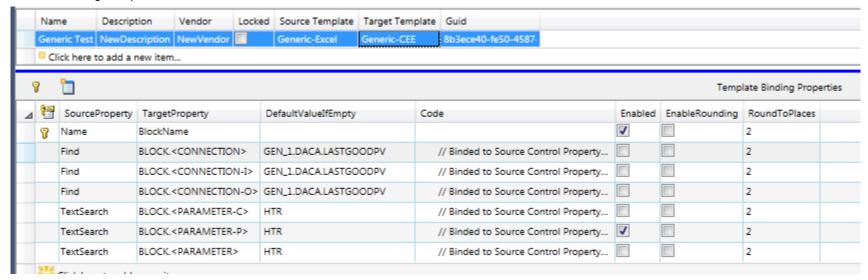
### Excel DataSource:



#### Control Property Template:



#### **Control Binding Template:**



## **Searching and Replacing Parameter Connection References**

You can search and replace Parameter Connection references by specifying the value of *BLOCK*.<*CONNECTION>* for the Target Control Property. You define this special Target Property in your Control Property Template as in the example above, which then allows you to select it for the Target Control Property in a Control Binding:

When searching and replacing Parameter Connection references you will define a column in your Excel DataSource such as "Find" that is used to specify the Parameter Connection reference to search for. The DefaultValuelfEmpty attribute for your Control Binding is used to specify the replacement Connection reference value. When this type of Control Binding usage is detected during generation of a Data Collection task all Control Module Parameter Connection references and expressions will be searched for the reference specified in the Control Binding Source property that defines the Excel Data Source column to use. In the example above the Control Binding source property is bound to the Excel Data Source column named *Find*. In the example above the replacement value is specified in the Control Binding *DefaultValuelfEmpty* attribute.

## Parameter references are searched in the following elements:

- Parameter reference connections
- Expression type parameters such used in AuxCalc and RegCalc type Function blocks.

## You can specify optional attributes in the Connection name format:

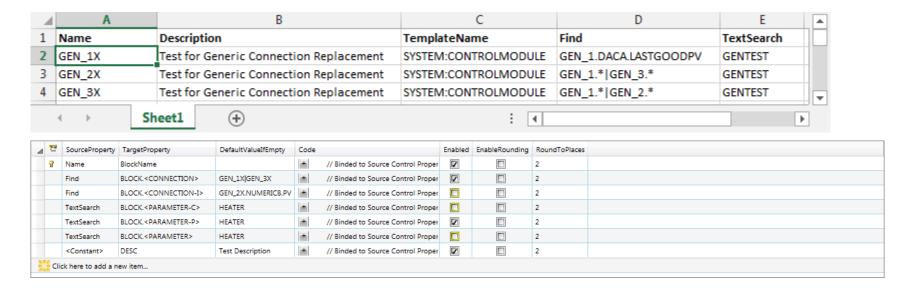
- BLOCK.<CONNECTION> indicates to search for both input and output Parameter Connection references.
- BLOCK.<CONNECTION-I> indicates to search for only input Parameter Connection references.
- BLOCK.<CONNECTION-O> indicates to search for only output Parameter Connection references. This would be for Block types such as PUSH.

## You can choose to find and replace using the wildcard character \* in the format CMname.\*:

- Specify the find reference as in GEN\_1.\* and the replacement value as specified in *DefaultValueIfEmpty* GEN\_1X. This will find all references to the Control Module "GEN\_1" regardless of the BlockName or Parameter Name path. Any reference to the Control Module "GEN 1" will be replaced with "GEN 1X".
  - a. Parameter references such as "GEN\_1.DACA.PV" would be replaced with "GEN\_1X.DACA.PV"
  - b. Expression references such as "GEN\_1.NUMERICA.PV > 10.0?~0.0: GEN\_1.NUMERICA.PV" would be replaced with "GEN\_1X.NUMERICA.PV > 10.0?~0.0: GEN\_1X.NUMERICA.PV".

You can choose to find and replace multiple values by separating search and find values with the '|' pipe character:

- Specify multiple find and replace values as in Find: "GEN\_1.\*|GEN\_2.\*" and Replace (DefaultValueIfEmpty): "GEN\_1X|GEN\_2X".
- Use this format to replace multiple references in a Control Module. You cannot specify multiple Control Bindings using the Target Property name BLOCK.<CONNECTION>. The Control Binding view enforces uniqueness in the Target Property names that are specified in the Control Binding Template.



## **Block Functions**

You can also execute custom functions such as deleting an embedded Function Block in a PKS Control Module. See <u>Control Binding</u> <u>Functions</u>.

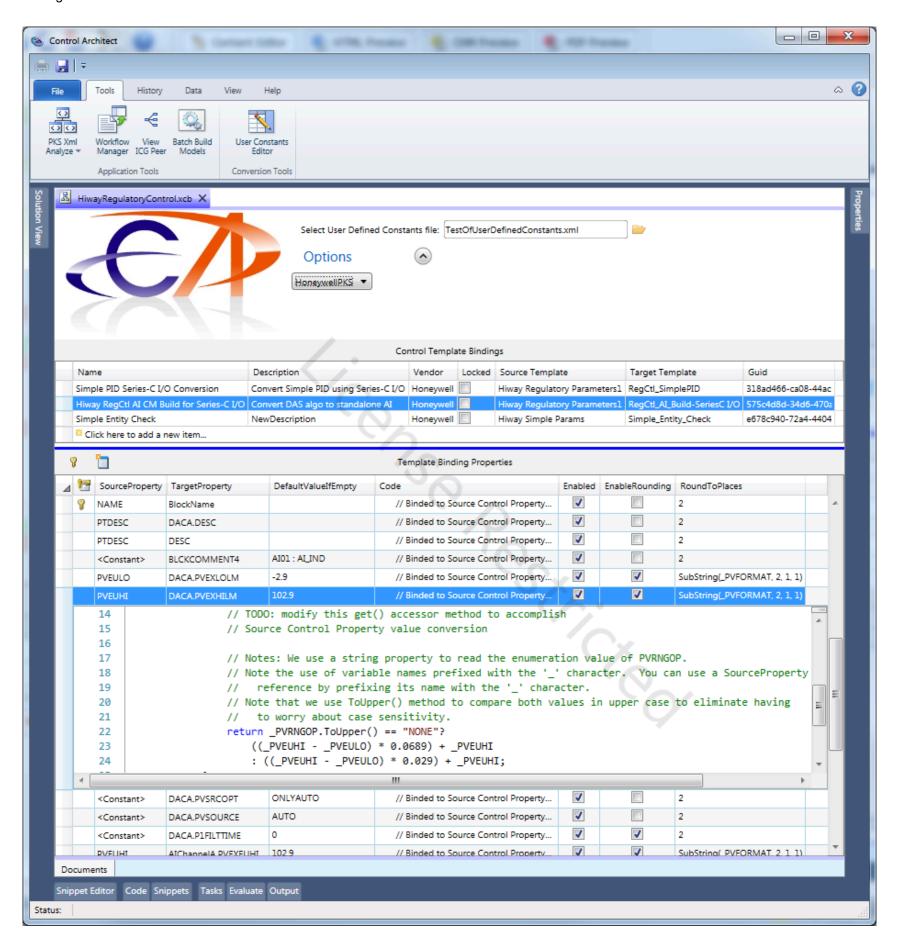
### **Tips & Tricks**

- You can save script code as a **Code Snippet** to build a library of commonly used expressions.
- You can use the key combination Ctrl+k, followed by the 'F' key to format a code snippet.
- You can restore the Default Value based on the value found in the BasedOn Template by pressing the **F9** key for a selected Control Binding item.
- You can modify a Control Binding script so that it ignores empty values in an Excel DataSource. Modify a Control Binding script such that the result value is its Target Control Property if the Excel Source Property value is empty.

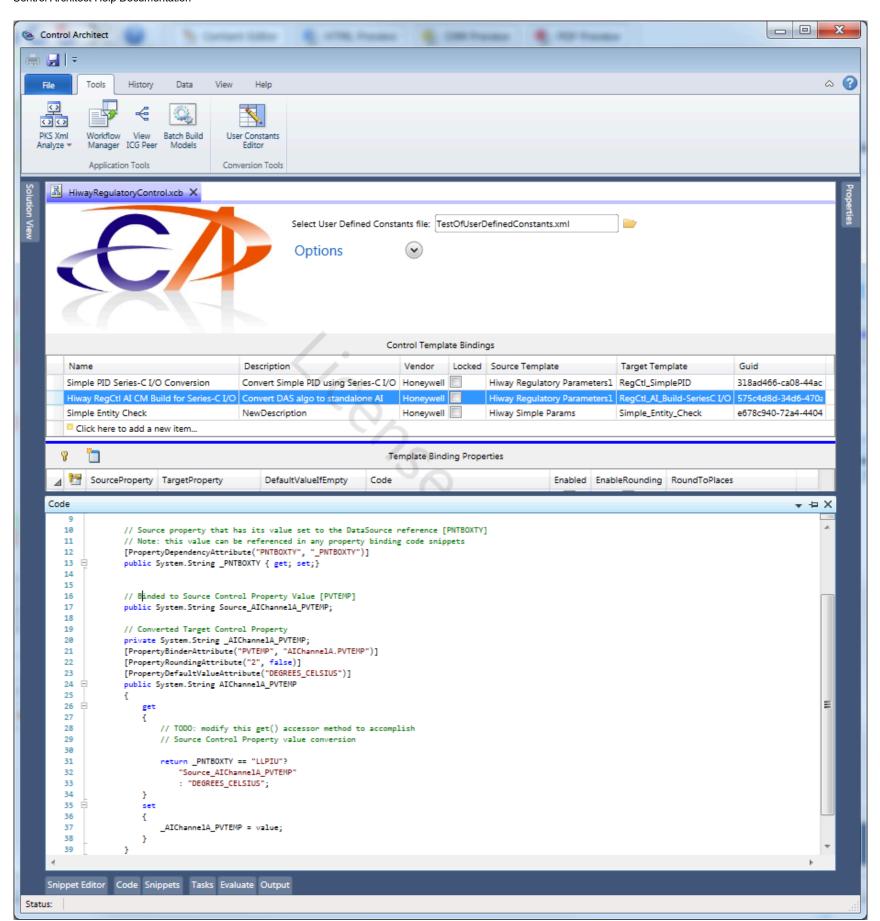
```
// Binded to Source Control Property Value [DACA EUDESC]
public System.String Source DACA EUDESC;
// Converted Target Control Property
private System.String DACA EUDESC;
[PropertyBinderAttribute("DACA.EUDESC", "DACA.EUDESC")]
[PropertyRoundingAttribute("2", false)]
[PropertyDefaultValueAttribute("")]
public System.String DACA_EUDESC
{
    get
    {
        // If the EUDESC value in the Excel DataSource is empty, then just return
        // the existing Entity Parameter value, which will result in no detection
        if (string.IsNullOrWhiteSpace(Source DACA EUDESC)) return DACA EUDESC;
        return Source DACA EUDESC;
    }
    set
        DACA EUDESC = value;
```

# **Control Binding Expression examples**

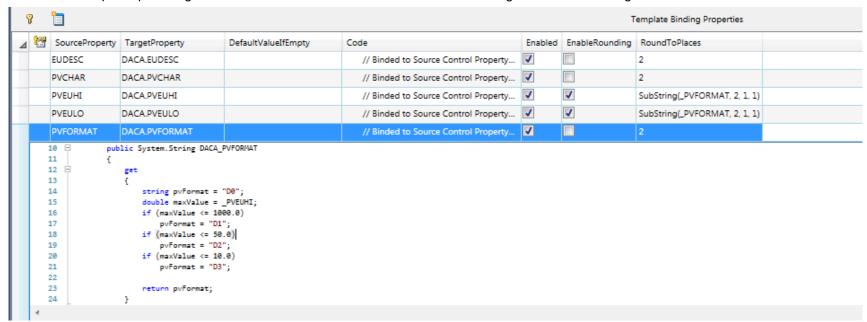
Below are some examples of expressions used when converting Honeywell Hiway Regulatory control entities to PKS Cxxx Experion control strategies.



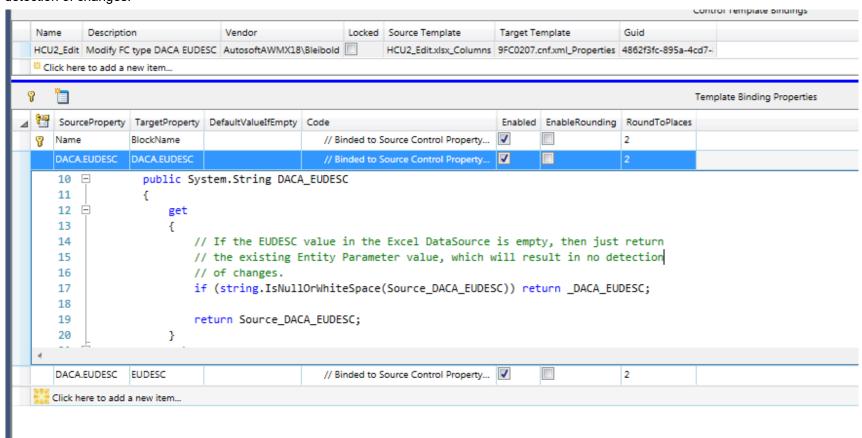
This is another example where the control binding expression code is viewed in the Code View after Compilation and Evaluation.



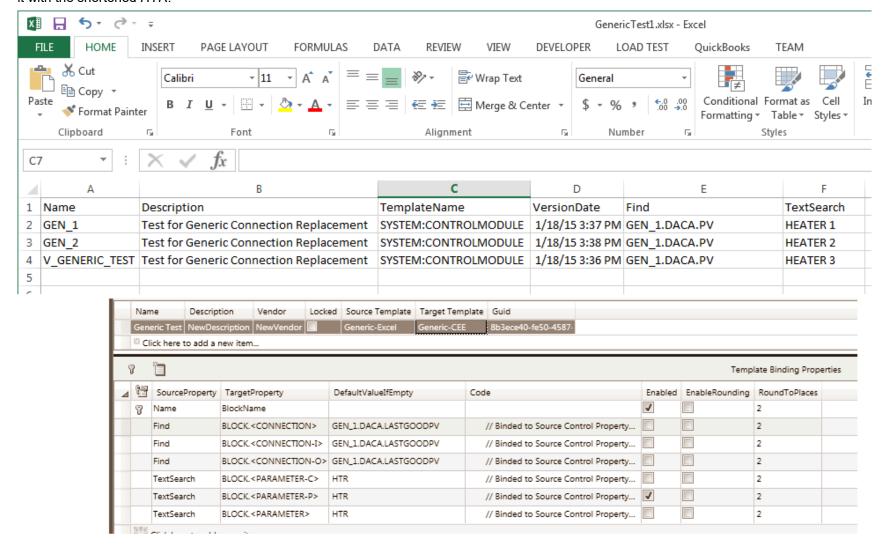
This is an example of providing a PVFORMAT enumeration value based on the configured maximum range value of the of the Function Block.



This is an example of scripting a Control Binding to ignore the DataSource value when it is empty by setting the result of the Control Binding script evaluation to its existing Parameter value, which the Data Collection will show both the existing and new values as identical resulting in no detection of changes.



This is an example of how to search and replace Parameter values and also Connection parameter references. Note how the Excel DataSource defines additional columns to contain the Connection reference and Parameter values to search for. The Control Binding *DefaultValueIfEmpty* attribute is used to define the replacement value. In this example we are replacing all uses of the Connection reference *GEN\_1.DACA.PV* with the new Connection reference *GEN\_1.DACA.LASTGOODPV*. We are also replacing references to the Parameter value *HEATER* and replacing it with the shortened *HTR*.



#### **Custom API Methods**

Control Architect provides custom API methods to facilitate Control Conversion tasks. The API methods can be used within a Control Binding C# generated script to facilitate common data retrieval and manipulation requirements.

#### **LookUp API Method**

This API Method allows users to search for data in another Worksheet in the same Excel Workbook being processed or in an external Excel Workbook file.

Prior to the development of this API method, users were entirely limited to using data within the Excel Worksheet that was currently being processed for conversion. Users often had to resort to using VLookup Excel functions to copy data from other Worksheets to the Worksheet with the primary data, which added time and complexity for the engineering task. This new LookUp(...) method eliminates this entirely.

This method is located in the Namespace: ControlConversion.Support.Core.Conversion

LookUp(string excelFileName, string excelWorksheetName, string excelColumnName, string primaryKeyLookupValue, string lookupColumnName)

#### Name Description

excelFileName The name of the Excel Workbook file that contains the data value to find.

excelWorksheetNameename of the Excel Worksheet within the Workbook file that contains the data value to find.

excelColumnNamehe name of the Worksheet Column that contains the lookup key to search for.

primaryKeyLookuptwaluelue to search for in the lookup key.

lookupColumnName name of the Worksheet column in the looked up data row to return its value from.

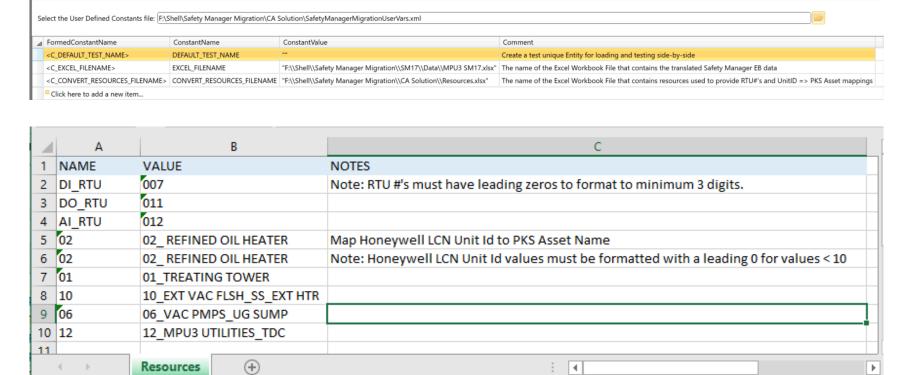
#### **Usage Examples**

This example below shows how to use the LookUp(...) method together with the use of user defined Custom variables. One of the LookUp methods below is referencing data from an Excel Workbook file as shown below.

This method call is referencing a lookup value in the Workbook file below:

var rtu = ControlConversion.Support.Core.Conversion.LookUp(\$<C\_CONVERT\_RESOURCES\_FILENAME>\$, "Resources", "NAME", "DI\_RTU", "VALUE");

The Custom Variable usage \$<C\_CONVERT\_RESOURCES\_FILENAME>\$ is defined to contain the file path to this Excel Workbook file as defined in the referenced User Defined Variable file:



Note the use of <code>Source\_PVSOURCE.Substring(0, Source\_PVSOURCE.IndexOf('.'))</code>, which takes the current value of the bound Source property <code>DISRC(1)</code> and extracts the connection reference Entity name component for the Lookup value. For example <code>DISCR(1)</code> would contain an EntityName.Parameter value as in <code>3HA3000.PVFL</code> where we need to extract the substring value <code>3HA3000</code> to lookup in the <code>[DIINNIM]</code> Worksheet and return the value in the column named <code>PLCADDR</code>.

This is the C# script that is generated when a Control Binding is established. We modified the default script code in the generated Property *get* accessor to return a value for the target Property name. The custom code starts with the comment line: // Custom code starts here and ends with the *get* accessor returning a value.

Copyright © 2024 by Automation Software Technology, Inc. All Rights Reserved.

```
// Binded to Source Control Property Value [DISRC(1)]
public System.String Source_PVSOURCE;
// Converted Target Control Property
private System.String PVSOURCE;
[PropertyBinderAttribute("DISRC(1)", "PVSOURCE")] \\
[PropertyRoundingAttribute("2", false)]
[PropertyDefaultValueAttribute("")]
[IsCMPropertyAttribute(false)]
public System.String PVSOURCE
      get
                 // Custom code starts here
                 int plcAddress = 0;
                 if (string.IsNullOrWhiteSpace(Source PVSOURCE) || Source PVSOURCE.IndexOf('.') == -1)
                      return string.Empty;
                  var stringAddress = ControlConversion.Support.Core.Conversion.LookUp($<C_EXCEL_FILENAME>$, "DIINNIM", "NAME",
             Source_PVSOURCE.Substring(0, Source_PVSOURCE.IndexOf('.')), "PLCADDR");
                  int.TryParse(stringAddress, out plcAddress);
                 var\ rtu = Control Conversion. Support. Core. Conversion. Look Up (\$<C\_CONVERT\_RESOURCES\_FILENAME>\$, "Resources", "NAME", to the control Conversion of the control Conversio
             "DI_RTU", "VALUE");
                 return rtu + ' ' + '0' + (plcAddress - 10000) + " 0";
       set
                PVSOURCE = value;
```

#### **ExistsInColumn API Method**

This API Method allows users to search and verify that a particular Column name exists in any Worksheet in the same Excel Workbook being processed or in an external Excel Workbook file.

Prior to the development of this API method, users were entirely limited to using data within the Excel Worksheet that was currently being processed for conversion.

This method is located in the Namespace: ControlConversion.Support.Core.Conversion

ExistsInColumn(string excelFileName, string excelWorksheetName, string searchColumnNameQuery)

#### Name Description

excelFileName The name of the Excel Workbook file that contains the data value to find.

excelWorksheetNameename of the Excel Worksheet within the Workbook file that contains the data value to find.

searchColumnNainteQuerry used to find the specified column name.

### **Usage Examples**

This example below shows how to use the ExistsInColumn(...) method together with the use of user defined Custom variables. One of the ExistsInColumn methods below is referencing data from an Excel Workbook file as shown below. This function was developed to allow users to determine if for example a referenced Entity name in a connection reference may exist in another point type table.

In the LCN realm when working with xPM and other controller type data exported in Honeywell Exception Build (EB) files we often have references on a Regulatory Control point type such as "CODSTN(1) = 45FY123.OP", which is typically the Analog Output Channel point that drives the final element in the field. This output reference could be an Analog Output point or it could be the setpoint being pushed as in "CODSTN(1) = 45FC124.SP". When building Templates for a simple PID strategy it will typically contain an Analog Input channel block, a PID function block, and an Analog Output channel block. During processing we can determine if this PID function is driving a hardwired AO Channel block, by parsing the connection reference "45FY123.OP" to get "45FY123" and search for this entity reference name in the "ANOUTNIM" Excel point table. If not found, then we can assume we don't want this Analog Output channel block in this PID strategy as it is driving another block.parameter type. So we can return the value "BLOCK\_DELETE" to notify the conversion process to remove this Analog Output channel block during processing.

Without the help of this function a user would have to provide this data manually embedded in the Worksheet point table that is being processed using various Excel function calls. Use of this function helps to minimize the amount of modification and pre-processing of the Excel data for use in a Control Conversion solution.

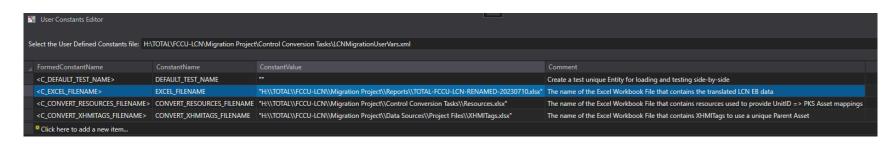
This method call is referencing a lookup value in the Workbook file below:

#### Control Architect Help Documentation

This string component in the function call above "NAME=' " + source1 + " ' " is the format for specifying a query when concatenating string values. If source1 has the value of 54FY123 then the query becomes "NAME='54FY123" where *NAME* is the name of the Excel worksheet (table) column to search for an Entity name reference in.

The return value is a Tuple<book, string> type where item1 (book) returns True if the specified column name exists, otherwise False; Item2 (string) returns an error string if item1 returns False.

The Custom Variable usage \$<C\_EXCEL\_FILENAME>\$ is defined to contain the file path to this Excel Workbook file as defined in the referenced User Defined Variable file:



This is the C# script that is generated when a Control Binding is established. We modified the default script code in the generated Property *get* accessor to return a value for the target Property name. The custom code starts with the comment line: // Custom code starts here and ends with the *get* accessor returning a value.

## **NonCee References**

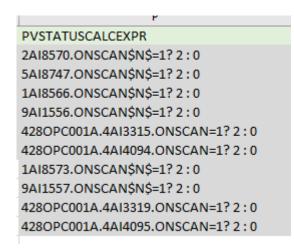
NonCee references in PKS Experion allow a Control Strategy executing in a CEE environment to reference parameter data in other node types such as LCN and SCADA. Typically in an ACE CEE host, we would use an OPCG for referencing other node types, but OPCG cannot be used in a C300 CEE strategy. Using a NonCee reference type allows the data to be treated as CDA.

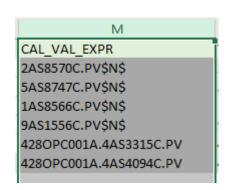
Control Builder supports adding what is called NonCee parameter references by simply specifying the EntityName.ParameterName reference as either a pin connection or reference within an AuxCalc expression. The caveat of this is that the Entity reference must be be resolvable to Control Builder at the time the reference is made. For Scada type references, this means the Scada entity must already be loaded to the Server and for LCN references the entity must also be loaded and reachable through TPNServer.

Control Modules when imported and exported contain special handling and identification for all NonCee references within blocks embedded in the Control Module. Control Architect generates this special identification for NonCee reference types, but requires the user to formally identify what references are NonCee types for them to be identified and generated as NonCee reference types. Control Architect supports both pin connection reference types and references within expressions typical of AuxCalc, EnAuxCalc, and RegCalc block types.

#### **Expression & Pin Connection NonCee References**

To specify that an Entity.Parameter reference in an expression or a connection is a NonCee type, the user will append "\$N\$" to the reference value, which will instruct the conversion execution to treat the Parameter reference as a NonCee type. The below Excel Data Source examples shows how to use this format when developing expressions that contain NonCee reference types. If this format is not applied, then the reference is treated as a normal CEE reference type.



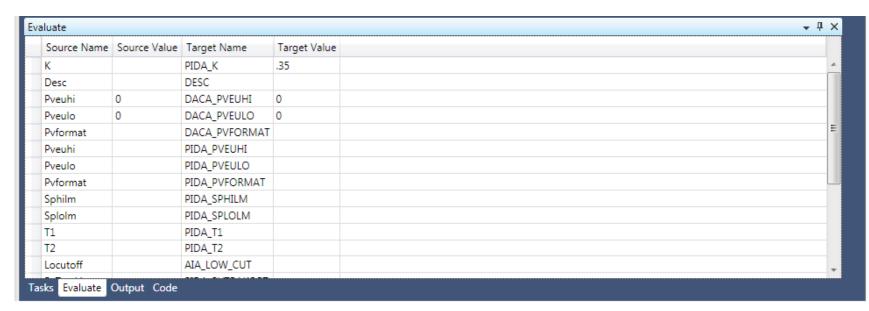


Example where connection references are made either to a NonCee Parameter reference or an OPC Gateway reference. Note that the format "\$N\$" is only applied as a suffix to the connection references that are not routed through an OPCG.



# **Control Binding Evaluation**

The Evaluation view allows the user to test code expressions that are defined in a set of control bindings.



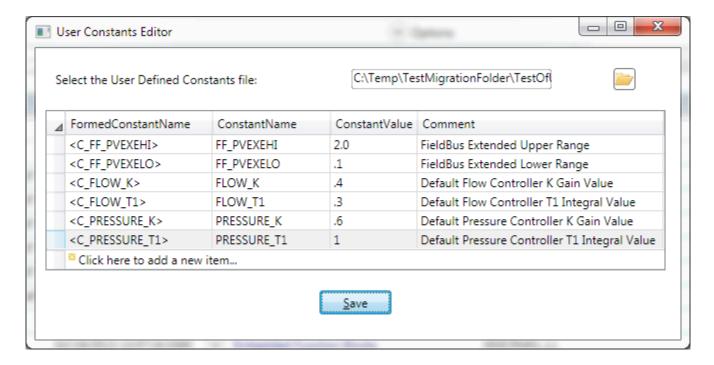
- When developing templates that define how source and target properties are bound a set of default code expressions are generated that provide the means for defining how a value from a source is converted into its required target value. By default, the generated code expression for a binding simply maps the value of the source Property to the target Property without any modification. Changes to the default mapping expressions can be tested using the *evaluation* function.
  - When the right mouse-click context menu item **Compile and Evaluate Property Snippet** is clicked, all selected control bindings are compiled. The results of the compilation that includes the generated expression source code class and its referenced control binding source and target properties are presented in the Evaluation view to allow the user to test the results of the code expressions developed for each control binding. Values can be entered in the column labeled "Source Value". The code expression will then be evaluated and its target Control Property value will be updated to reflect the code expression result as indicated in the column labeled "Target Value".
  - If source or target Control Property of a control binding is changed or its code expression is modified it must be recompiled to reflect the change to the Evaluation view.
  - To evaluate all control bindings click in the Properties grid of the Control Binding view and press the key combination **Ctrl+A** to select all control binding items. Right mouse-click to display the context menu and select the menu item **Compile and Evaluate Property Snippet.** Only the items selected will be compiled and presented in the Evaluation view. Any existing items that were present in the view will be cleared and replaced by the results of the new compilation.

#### **Features**

- Target properties that are dependent on a particular source property will automatically be updated when the value for the source Property is changed.
- Value rounding options are evaluated and applied when testing expression bindings.

#### **Constants Editor**

The User Constants editor view allows the user to define a global set of constant variables that can then be referenced in Control Bindings.



Typically a project will define a set of standards and rules that define how particular parameter types and their values will conform to a new control platform during a software migration. Examples would include defaulting FieldBus type extended ranges vs. standard Series C type I/O. Defaulting tuning parameters based on the type of process controller such as flow, pressure, temperature, level, etc.

- The constant alias that you define in this editor will become automatically formed using the format <C\_*UserName*>.
- Duplicate constant names are not allowed.
- · Multiple constant files are supported.
- When defining Control Binding relationships in a <u>Control Binding view</u>, a constant file can be referenced to allow its contained constant definitions to be used in a Control binding.
- The Constant file that is saved is a standard Xml formatted file type that can be shared with other users.

#### Access

Access this tool through the Ribbon "Tools" tab group.

## **Custom Variables**

Control Architect defines a set of predefined variables that can be used in a **<Constant>** type source property binding. A custom variable is specified in the format of *\$(VariableName)* where VariableName is a name that is predefined.

## **Custom Predefined Variable Names**

The following list of predefined variable names can be used as the *default* value in a **Constant>** type source property binding:

#### **ControlBinding Template References**

- CBT\_NAME The name of the Control Binding template.
- CBT\_DESCRIPTION The description for the Control Binding template.
- CBT\_VENDOR The vendor associated with the Control Binding template.
- CBT\_SOURCETEMPLATE The Control Property template name that provides the set of properties used in the Source Property of the binding.
- CBT\_TARGETTEMPLATE The Control Property template name that provides the set of properties used in the Target Property of the binding.
- CBT\_SOURCEFILE The Control Property template file name that provides the set of properties used in the Source Property of the binding.
- CBT\_TARGETFILE The Control Property template file name that provides the set of properties used in the Target Property of the binding.
- CBP\_SOURCEPROPERTYNAME The Source Property used in the control binding.
- CBP\_TARGETPROPERTYNAME The Target Property used in the control binding.

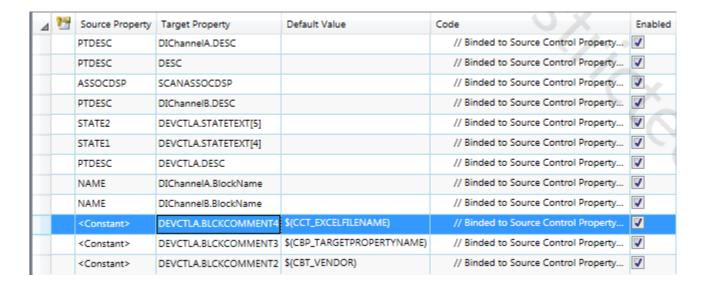
#### **ControlConversion Template References**

- CCT EXCELFILENAME The Excel file name that is used as the Data Source for this Conversion template.
- CCA BASEDONFILE The PKS Control Module XML configuration file that is used as the basis for Bulk building and editing.
- CCA\_TASKNAME The Control Conversion Task Name.
- CCA FILTEREXPRESSION The filter expression used to filter the Data Source view.
- CCA\_DATASOURCETABLE The Data Source table that contains the source entities.

#### **Notes**

When using a Custom Variable in a Control binding, specify the variable in the Default value of the binding in the format of  $(CCT\_EXCELFILENAME)$ . The variable name must be enclosed within the characters (...).

The example below shows 3 constant type property bindings that when evaluated will retrieve the content of the specified custom predefined variable and assign it to the target property value.



# **Code Snippets**

Control Architect provides the ability to define and save often used pieces of C# code used to define Control Binding expressions as *Code Snippets*. Code snippets provide a library of samples of C# scripting that can be easily dragged and dropped onto the script editor.

# **Creating a Code Snippet**

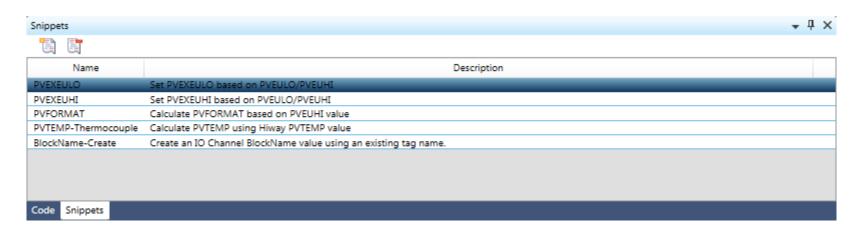
A Code Snippet can be created in one of two ways as follows:

- 1. Click the Add-New Snippet Button in the Code Snippet view.
- 2. From the Script Editor surface of a Control Binding, select a text range and left mouse-click and hold the button while dragging the selected text range to the Code Snippet View, then release the mouse button to drop the selected text.

## **Using a Code Snippet**

To use an existing Code Snippet, left mouse-click to select the Code Snippet in the Code Snippet view and drag it to the text position in the script editor surface, then release the mouse button to drop the dragged snippet text.

You can also select the script code using Ctrl+A and Ctrl+C to copy it to the Windows Clipboard. Use Ctrl+V to paste the code script in the editor.



## **Edit Existing Code Snippets**

You can edit existing code snippets by selecting the Code Snippet to edit in the Code Snippet view. This will update the Code Snippet Editor view with the contents of the selected code snippet for editing. Any changes made in the Code Snippet Editor view will automatically be saved.



## **Delete an existing Code Snippet**

A Code Snippet can be deleted by clicking the Remove Code Snippet button to delete the selected Code Snippet item.

### **Application Option to Auto Activate the Code Snippet Editor View**

You can turn on/off the <u>application option</u> to automatically activate the Code Snippet Editor view when a code snippet item is selected in the Code Snippet view.

## Notes

- Your code snippets are saved to an json format file that is located in your local application data folder. This can typically be located at C:\Users\<user-name>\AppData\Local\Automation Software Technology, Inc.\Control Architect\UserSnippets.json where AppData is the hidden folder.
- You can copy this file to save as a backup or share with other users.
- This file can be edited off-line using a JSON editor. When edited, Control Architect must be restarted to reread the file.
- Hover your mouse cursor over a Snippet item in the Snippet View to show a ToolTip that show the Snippet contents.

Name	D	escription	***************************************
PVEXEULO	Set PVEXEULO based on PVEULO/PVEUHI		
PVEXEUHI	Set PVEXEUHI based on PVEULO/PVEUHI		
PVFORMAT	Calculate PVFORMAT based on PVEUHI value		
PVTEMP-Thermocouple	Calculate PVTEMP using Hiway PVTEMP value		
BlockName-Create	Bini	1	
	Description: Calculate PVFORMAT based on PVEUHI value		
	Calculate FVI ONIVIAT based off FVEOTIL value		
	Snippet:		
	string pvFormat = "D0";		
	double maxValue = _PVEUHI;		
	if (maxValue <= 1000.0)		
	pvFormat = "D1";		
	if (maxValue <= 50.0)		
	pvFormat = "D2";		
	if (maxValue <= 10.0)		
	pvFormat = "D3";		
	return pvFormat;		
	recom pyronnac,		
***************************************			

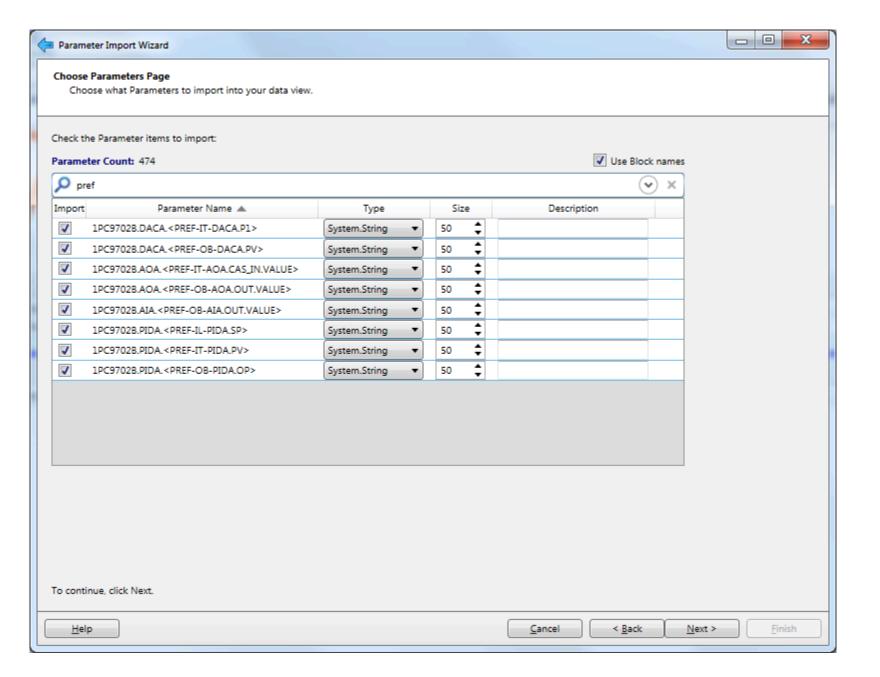
# **Binding to Block Pin connections**

Defining a function block Pin connection as a Property to reference in a Control Binding is supported in the Control Property Template. When importing properties from a typical Cxxx/ACE type XML configuration file all exposed Pin connections in the referenced XML configuration file will be available for selection in the Parameter list.

Parameter Pin connections are formatted as a Property type using the following notation:

- <PREF-LL-BBBB.PPPP> where
  - a. PREF defines this type of Property (parameter) as a Pin connection type
  - b. LL defines the location of the connection Pin in relation to its Function Block surface
    - i. IT = InputTop
    - ii. IL = InputLeft
    - iii. OB = OutputBottom
    - iv. OR = OutputRight
  - c. BBBB defines the Function Block name that contains the Pin connection reference
  - d. PPPP defines the Parameter that the Pin connection references on the Function Block

By default Pin connection parameters are typed as a *String* type.



# **Block Building**

#### **Overview**

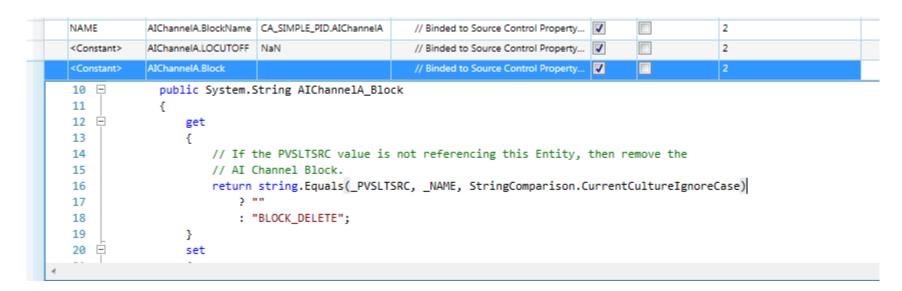
The Control Binding Template supports the use of constants and expressions to define the removal of blocks from a Control Strategy. This version of Control Architect only supports the removal of existing blocks that form a default Control Strategy or Module.

This functionality allows you to define a common Template for a Control Conversion task that during execution will allow you to programmatically remove functions or blocks that don't apply. For example you can define a simple PID Control Strategy that contains an Analog Input, Regulatory PID function, and an Analog Output. If during processing you find that the new Control Strategy does not require an Analog Input/Output Block, then you can choose to specify in the Binding expression that this Analog Input/Output Block is to be removed. This saves a lot of effort in having to maintain multiple Template variations to account for the possibilities that an Analog Input/Output function may or may not be required.

#### **Block Removal**

You can define in a Control Binding an expression that will determine if a particular Template Block should be removed. This type of Control Binding requires that the Source Control Property is a type of constant *Constant>* and the Target Control Property must be the name of the Function Block appended with the "Block" value as in "AlChannelA.Block". You must manually add this new Control Property "AlChannelA.Block" to the CM Template whose block or blocks may require deletion. The "Block" name is case sensitive, which means if you specify "BLOCK" it will fail. You would develop an expression for this binding that can determine a valid command value to apply to the binding when executed. If the returned command value is an empty string as in "" or string.Empty, then the Delete Block command will not be executed.

The Block command value for removal must be set to the value "BLOCK\_DELETE" in order for the Control Conversion process to recognize and execute the removal command.



You can also add a Control Binding where the source Property name is the *Constant>* and the target Control Property is a user added Control Property that is the name of the embedded Function Block appended with the text '.Block' as in "AlChannelA.Block" where "AlChannelA" is the name of the embedded Function Block.



When a Data Collection is executed with a reference to this Control Binding, a message will indicate that the Block will be deleted.

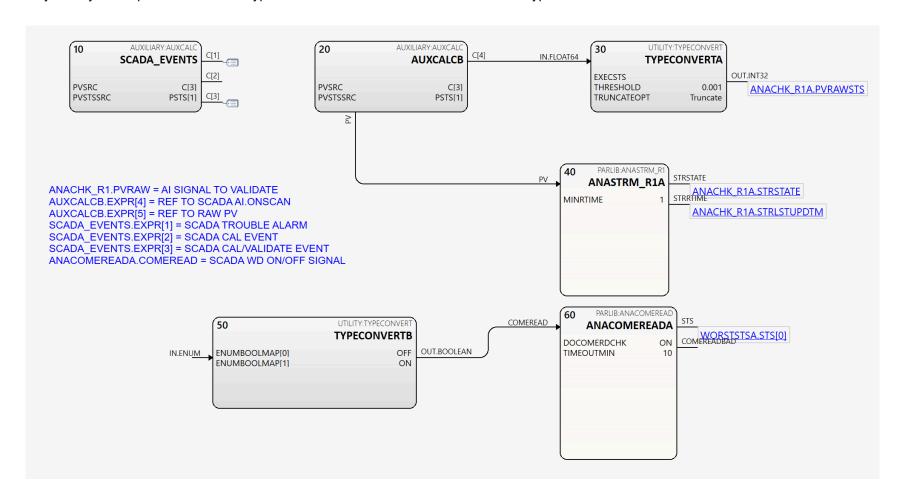


# **Disabling Bindings**

Control Architect supports the programmatic disabling of a Control Binding by using a custom command value *REMOVEBINDING* that is returned in the Getter method of a Control Binding script property.

Examples of how or where users may apply this type of function would include adding Parameter References that are based on the Strategy host { ACE, C300 } where either a NonCee parameter reference or a CEE parameter reference is to be applied, where both connection type references can be configured as 2 Control Bindings, but only one or the other is executed based on conditions specified in the Property script.

Here is an example of coding a Property script for the above scenario. The generated Control Modules may either be loaded into an ACE or C300. Strategies loaded into an ACE Server that have connections to Scada data will typically use OPCG gateways, while strategies loaded into a C300 will use NonCee references as OPCG gateways are not accessible from a C300. Depending on the host CEE the Parameter Reference connections will either be routing using an OPCG (ACE host) or be routed directly as a NonCee Parameter Reference type, which may or may not require an additional TypeConverter Function block for Scada Status types.



The above Chart View represents part of a Control Module whose Blocks are being connected during Control Conversion using the Control Binding set below. The Control Bindings of interest below specify whether a Parameter Reference connection is established on the TYPECONVERTB.IN.ENUM pin or the ANACOMEREADA.COMEREAD input pin. If deciding to pin to the ANACOMEREADA.COMEREAD input pin, then the TYPECONVERTB block must be removed during conversion execution.

4	SourceProperty	TargetProperty	DefaultValueIfEmpty	Code		Enabled	EnableRounding	RoundToPlaces
	PVFORMAT	DACA.PVFORMAT	D1	+	// Binded to Source Control.			2
	SCANASSOCDSP	SCANASSOCDSP		+	// Binded to Source Control.			2
	MAYBEBADTMOM	ANACHK_R1A.MAYBEBADTMOM	10	+	// Binded to Source Control.			2
	STRUPDTMOM	ANACHK_R1A.STRUPDTMOM	10	+	// Binded to Source Control.			2
	DESC	ANACHK_R1A.DESC		+	// Binded to Source Control.			2
	TROUBLEEXPR	SCADA_EVENTS.EXPR[1]		+	// Binded to Source Control.			2
	CALEXPR	SCADA_EVENTS.EXPR[2]		+	// Binded to Source Control.			2
	CAL_VAL_EXPR	SCADA_EVENTS.EXPR[3]		+	// Binded to Source Control.			2
	WATCHDOG	TYPECONVERTB. <pref-il-typeconvertb.in.enum></pref-il-typeconvertb.in.enum>		+	// Binded to Source Control.			2
	PVEULO	ANACHK_R1A.ANALOLIM	0	+	// Binded to Source Control.			2
	PVEUHI	ANACHK_R1A.ANAHILIM	0	+	// Binded to Source Control.			2
	TIMEOUTMIN	ANACOMEREADA.TIMEOUTMIN	10	+	// Binded to Source Control.			2
	<constant></constant>	TYPECONVERTB.Block		+	// Binded to Source Control.			2
	WATCHDOG	ANACOMEREADA. < PREF-IL-ANACOMEREADA. COMEREAD >		+	// Binded to Source Control.			2

The above snapshot shows a set of Control Bindings for generating an Analyzer Validation CM where the 3 highlighted Control Bindings use either a normal OPCG gateway for the Scada point reference or a C300 NonCee reference path to the Scada point. The TargetProperty *TYPECONVERTB.Block* is being used to programmatically remove the *TypeConvert* Function block if the host CEE is an ACE vs C300 where no type conversion would be required if the source Scada reference is using an OPCG gateway.

```
WATCHDOG
                 TYPECONVERTB. < PREF-IL-TYPECONVERTB.IN.ENUM >
             [IsCMPropertyAttribute(false)]
10
             public System.String TYPECONVERTB_PREF_IL_TYPECONVERTB_IN_ENUM
11 🗏
12
13
14
                     var stringValue = ControlConversion.Support.Core.Conversion.LookUp($<C_EXCEL_RESOURCES_FILENAME>$, "RESOURCES", "UNIT", _FLAREID, "HostInC300");
15
                     var isHostedInC300 = string.Compare(stringValue, "YES", StringComparison.CurrentCultureIgnoreCase) == 0;
16
                     return !isHostedInC300
17
                           "<REMOVEBINDING>
18
                         : Source_TYPECONVERTB_PREF_IL_TYPECONVERTB_IN_ENUM;
19
                 }
```

The above Control Binding script shows an example of how conditions are used to determine if this Control Binding will be processed during the Conversion execution. The Control Binding property will return the command REMOVEBINDING if the generated Control Module will be loaded into an ACE CEE.

```
WATCHDOG
                                                                                             // Binded to Source Control
                 ANACOMEREADA. < PREF-IL-ANACOMEREADA. COMEREAD >
11
             public System.String ANACOMEREADA_PREF_IL_ANACOMEREADA_COMEREAD
12
             {
13
14
                      var stringValue = ControlConversion.Support.Core.Conversion.LookUp($<C_EXCEL_RESOURCES_FILENAME>$, "RESOURCES", "UNIT", _FLAREID, "HostInC300");
15
                      var isHostedInC300 = string.Compare(stringValue, "YES", StringComparison.CurrentCultureIgnoreCase) == 0;
16
17
                      return isHostedInC300
18
                          ? "<REMOVEBINDING>"
19
                          : Source_ANACOMEREADA_PREF_IL_ANACOMEREADA_COMEREAD;
20
```

The above Control Binding script shows an example of how conditions are used to determine if this Control Binding will be processed during the Conversion execution. The Control Binding property will return the command <REMOVEBINDING> if the generated Control Module will be loaded into a C300 CEE as it will use a NonCee reference that will require a type conversion.

```
// Binded to Source Control.
<Constant>
                  TYPECONVERTB.Block
             public System.String TYPECONVERTB_Block
             {
13
                      var stringValue = ControlConversion.Support.Core.Conversion.LookUp($<C_EXCEL_RESOURCES_FILENAME>$, "RESOURCES", "UNIT", _FLAREID, "HostInC300");
15
                      var isHostedInC300 = string.Compare(stringValue, "YES", StringComparison.CurrentCultureIgnoreCase) == 0;
16
17
                      return isHostedInC300
                          ? string.Empty
18
                          : "BLOCK DELETE";
19
20
21
                  set
```

The above Control Binding is used to programmatically remove the *TYPECONVERTB* Block if the host CEE for this CM will be an ACE. You must manually add this new Control Property "TYPECONVERTB.Block" to the CM Template whose block or blocks may require deletion. The "Block" name is case sensitive, which means if you specify "BLOCK" it will fail. Returning the command text *BLOCK\_DELETE* will instruct the conversion execution to remove the Block with a BlockName of "TYPECONVERTB".

```
Note the use of the API method LookUp(...):
```

```
ControlConversion.Support.Core.Conversion.LookUp($<C_EXCEL_RESOURCES_FILENAME>$, "RESOURCES", "UNIT", _FLAREID, "HostInC300");
```

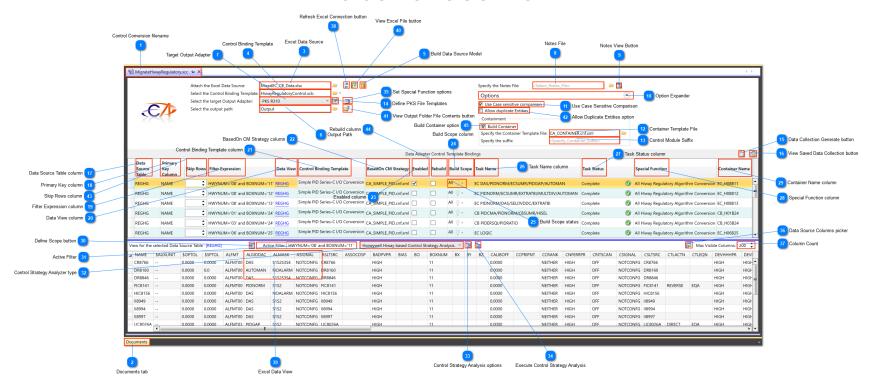
This API method allows users to provide common resources in a separate Excel Workbook or Worksheet that can be queried using the data as indicated in this example Excel Worksheet. The above Control Binding scripts determine if the host CEE is either an ACE or C300 based on the "FLAREID" value for the entity.

4	Α	В
1	UNIT	HostInC300
2	VPS5	Yes
3	DCU2	Yes
4	HCU2	Yes
5	SBU2	Yes
6	FCCU	No
7	ALKY	No
Q		

## **Control Conversion View**

The Control Conversion View is where bindings between an Excel Data Source file and a Control Binding template is established. The Control Binding template that establishes a binding between named Data Source column names and target control system type parameter names is bound to a file based Excel Data Source that will provide the data that will be used in a Control Conversion task such as <u>Bulk Building Cxxx</u> configuration XML files.

#### **Control Conversion View**



Control Conversion filename



The name of the Control Conversion Template instance. Hover your mouse cursor over this tab to view its full file path in a ToolTip.

Documents tab

# Documents

Control Template views are contained within this Documents tab well.

Excel Data Source

# MixedEC\_CB\_Data.xlsx

Click the Browse button to select an Excel File that contains the source of the data to convert. Control Conversion only supports Excel as the Data Source type.

Control Binding Template

# HiwayRegulatoryControl.xcb

Click the drop-down button to select a Control Binding Template file to use as the source of Control Bindings. Clicking the Browse button will display a list of all Control Binding type (\*.xcb) files that reside in the loaded Solution. This template defines the control bindings that will be used to bind the data from the Excel DataSource to the target control strategy parameter data.

Build Data Source Model



Click this button to build a data Model using the attached Excel Data Source. The Data Source Model provides Control Strategy data and references used when analyzing Control strategies for displaying a Control Graph.

Output Path

Output

Click the Browse button to select a folder in your file system where your Control Conversion output files will reside.

Target Output Adapter



Select from the drop-down list the type of output data that will be generated using this Control Conversion Template.

**Notes File** 

<Select\_Notes\_File>

Click the Browse button to select a document file to attach to this Control Conversion Template as a reference. You can open the document using its registered Open Verb by clicking the View button.

Notes View Button



Click this button to open the attached document reference. The registered Open Verb is used to open the file. For example if the document is an Excel file, then the Open Verb should be registered to use Excel as the application to open the file.

Option Expander

# **Options**

Click this expander button to expand/collapse the options view.

**Use Case Sensitive Comparison** 



Use Case sensitive comparison

Choose this option to automatically filter the Data Collection Parameter value comparison results based on case sensitivity.

Container Template File

# CA CONTAINER.cnf.xml

Click the Browse button to select a PKS XML Configuration file to use as an empty Control Module. This Control Module should not contain any embedded Function blocks. This function supports hierarchical embedded Control Module development.

Control Module Suffix

<Specify\_Container\_Suffix>

Specify the suffix to append to all auto-generated container Control Module strategies. The container Control Module that is generated is used to contain (embed) Control strategies that are generated as a result of executing a Data Collection saved from this view in a Workflow Control Conversion Activity task. This is currently not supported.

## Define PKS File Templates



Click this button to define the mapping between Honeywell PKS XML configuration templates and Control Architect defined identifiers. Control Architect provides a solution for migrating Honeywell legacy Control System platforms to EPKS that require PKS XML configuration templates that contain a specific Function Block structure. During the conversion analysis the default BasedOn type may need to be changed to properly build the converted Control Strategy. The conversion analyzer will define a new BasedOn Template that is predefined in this set of mappings. This button is only visible if a PKS type is chosen as the Output Adapter type.

#### Data Collection Generate button



Click this button to generate a Data Collection for the enabled Data Adapter Control Template bindings.

# View Saved Data Collection button



Click this button to view the last saved Data Collection view results. When a Data Collection is generated you must save the results that are then used in a Workflow Control Conversion Activity task. When the Control Conversion Activity task has been executed, you can view the results of this activity in this saved Data Collection. The status of all Parameter item conversions are automatically updated in the Data Collection.

## Data Source Table column

# Data

# Source

# **Table**

Select from the drop-down list a Data Table to use as the source of the data. Each Worksheet found in the attached Excel Workbook file is identified as a Table in this list. You can specify multiple control conversion bindings in this DataGrid where each DataGrid row represents a binding between a single DataSource table (Excel Worksheet) and a target set of Control properties (parameters) that represent a Control Strategy.

18

## **Primary Key column**

# **Primary**

# Key

# Column

Select the data column in this Data Table that represents the Primary Key to use as the data that uniquely identifies each entity in the data. This will typically be a data column that identifies the Control Strategy or Entity name.

19

## Filter Expression column

# Filter Expression

Specify an <u>expression</u> that is used to filter the data within the selected Data Table. You would typically use this to filter a scope of strategies that will conform using the adapter specified Control Binding Template file and its corresponding BasedOn file.

20

## **Data View column**

## Data View

Click this Hyperlink to view the Data Source table data for the specified Data Table.

21

### Control Binding Template column

# **Control Binding Template**

Select the Control Binding Template that will be used that defines the binding or mapping between source data and the target Control Strategy parameters. The Control Binding template file that is attached supports multiple control binding types.

22

### BasedOn CM Strategy column

# BasedOn CM Strategy

Click here to change the BasedOn Template file that will be used as the basis for all output files that are created. This file displayed here is the file that was used to import the Control properties for the specified Target Control Property source in the referenced Control Binding Template. You can click within this data cell to display a Browse button that will allow you to change the file reference.

23

## **Enabled column**

# **Enabled**

Check this box to enable this Adapter binding for processing when a Data Collection is executed.

Copyright © 2024 by Automation Software Technology, Inc. All Rights Reserved.



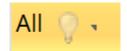
**Build Scope column** 

# **Build Scope**

By default all Data Source entities visible in the Data Source view become the scope when executing the Data Collection. You can choose to define this scope by selecting one more more Data Source rows in the Data Source view and clicking the <a href="Define Scope">Define Scope</a> button.



## **Build Scope status**



This status reflects the current scope of Data Source entities to use when the Data Collection task is executed. By default all entities contained in the selected Data Source Table are defined as the scope. If you choose to redefine this scope, you can click the drop-down button in this cell to display the current scope list of entities. You can also remove entities from this list by selecting them and clicking the **Delete** key to remove them.



## Task Name column

# Task Name

Provide a brief description of this Adapter binding that will be used to identify its usage when generating Data Collections.

27

## Task Status column

# **Task Status**

Displays the current status of this Adapter binding when used in a Data Collection task.



### Special Function column

# **Special Function**

Select from the drop-list a Special Function that is to be the basis for the Conversion analysis. Control Architect provides several special function types that provide support for legacy Honeywell Control System platform conversions.



## **Container Name column**

# **Container Name**

Specify a Container name here that will be applied to all Control modules that are generated. This provides support for hierarchical Control Module development. When a name is specified here, the Data Collection task will automatically parent all generated Control modules to this Container name. The containing Control Module that is also generated will use the PKS XML configuration file specified in the options. You can specify a hierarchy of embedded Control modules to create as in "HIWAY01\EC\_BOX32". A task with a hierarchical specification such as this example would embed the Control Module in the container named "EC\_BOX32" would itself be embedded in the container Module "HIWAY01".



# **Define Scope button**



Click this button to define the currently selected Data Source items as the scope of the Data Collection for the Adapter binding.

# 31

## **Active Filter**

# Active Filter: HWYNUM='08' and BOXNUM='11'

Displays the current filter expression that is active for the Data Source view. By default when the <u>Data View Hyperlink</u> is clicked to view its Table data, the expression that is specified in the Adapter binding <u>Filter Expression column</u> is used as the filter expression source. You can also change this expression here to change the filter temporarily without changing the Adapter binding filter expression.

32

# **Control Strategy Analyzer type**

# Honeywell Hiway based Control Strategy Analysis. \*

Select the type of analysis that is to be executed on the current Data Source view.

33

# **Control Strategy Analysis options**



Click this button to specify the options to use when analyzing the Data Source data.

34

# **Execute Control Strategy Analysis**



Click this button to generate a Control Graph based on analysis of the selected Data Source Entity row.

35

## **Set Special Function options**



Click this button to set the options for the selected Special Function type.

36

# **Data Source Columns picker**



Click this button to choose the columns that are displayed in the Data Source view. By default, the first 100 column found in the Data Source Table are displayed to improve performance when working with the Data Source view. Some Data Source tables such as a table that holds Honeywell Application Module Regulatory types may contain a 1000 or more columns depending on the size of any referenced CL Package items.

37

### **Column Count**



Specify the number of columns to view in the Data Source View. Data Source tables such as [CUSTOMAM] may contain hundreds of columns that can impact the performance of the DataGrid View that displays the Data Source table data. It is recommended to keep the number of visible columns in this view < 200. By default the view will pick the specified column count in alphabetic order. In special cases where you need to view more parameters click the Columns picker and check the additional columns to view.

When analyzing data to display in the Control Graph View it is required that certain columns be visible in order to properly graph data connections. For example in an Extended Controller the columns ZSLTSRC and ZSIGNAL must be visible to be able to read the data references specified in these data columns for display in the Control Graph View.

38

## Refresh Excel Connection button



Click this button to refresh the Excel Data Source connection. When you make changes to the referenced Excel file while this view is opened, click this button to refresh the connection to be able to see the changed data.

39

# **Excel Data View**

ALGIDDAC	ALMASK	ASIGNAL	,
DAS	S1S2S3S4	NOTCONFG	(
AUTOMAN	NOALARM	NOTCONFG	C
DAS	S1S2S3S4	NOTCONFG	[

When you click a hyperlink Data table reference in the <u>Data View column</u> its Excel data is displayed in this view.

View Excel File button



Click this button to launch Excel with the attached Excel Workbook file.

View Output Folder File Contents button



Click this button to view the Output folder in File Explorer.

Allow Duplicate Entities option

Allow duplicate Entities

Choose this option to support an Excel document where Entity names are duplicated for example in the NAME column. This allows you to develop Excel Data sources where Parameters are defined on an Excel row basis vs. column basis.

👧 Skip Rows column

**Skip Rows** 

Specify the number of Excel data rows to skip beginning with the1st row. Use this to skip unwanted data that is often embedded in the beginning of an Excel document.

Rebuild column

Rebuild

Choose this option when generating a Data Collection to rebuild the output entities and not treat existing output entities as being modified. This is a one-shot option and will be reset after executing a Data Collection.

You will often want to use this option if the target set of properties has changed, which may invalidate the existing output entities as new Parameters may need to be added or the structure of the configuration document has change such as adding/deleted an embedded block.

Build Container option

Build Container

Choose this option to auto-build a containing Module for entities that are generated. The specified Control Module container Template file along with the specified suffix are used to generate the new Control Module entity.

# **Defining Control Conversion Bindings**

Building a Control Conversion binding requires binding predefined Control Binding Templates to a file based Excel DataSource.

- 1. Select the Excel file that will be the source of the data used in a Control Conversion Task
- 2. Select the Control Binding template that defines the bindings between DataSource column data and target control system parameters.

## Define the source and target of a Control Conversion binding

The selected Excel DataSource and Control Binding template is used to define a Control Conversion binding between a DataSource table (Excel Worksheet) and the target DataSource such as an Experion Control Module or an XPM Entity.

Excel DataSource files may contain multiple sheets of data where each Excel Worksheet represents a data table. Typically Excel Worksheets would be developed based on a particular Template or Control Strategy such as a sheet named [ANLINHG] that contains all Hiway based Analog Input entity data, [REGHG] that contains all Hiway based Regulatory entity data, etc. used for Hiway migration.

For more general control strategy building, you might have an Excel Worksheet named [SimplePID] that contains all the parameter data required to configure a simple PID type control strategy including the Process Measurement variable parameters and Control Device variable parameters used to establish the configuration for the Analog Input and Analog Output functions.

To define the Control Conversion binding:

- 1. Select the DataSource table from the attached Excel file DataSource
- 2. Select the column of this DataSource table to be used as the Primary Key. This primary key defines the data in the DataSource table that uniquely identifies it, which should always be the Entity name. This primary key also establishes the indexing and sorting for this data table.
  - 1. Provide a filter expression to query for a subset of data if required. If the data table specified contains a large data set where multiple data Hiway's and boxes are present, then you might need to filter the data based on a particular HWYNUM or BOXNUM value or in the case of Hiway Regulatory Control data, you may choose to filter the data table data based on a particular Regulatory Control algorithm. If for example you choose to filter a data set based on a column named BOXNUM, you would enter an expression in the form of BOXNUM=27. The DataSource table view would only contain the data rows that satisfy the filter condition BOXNUM=27.
  - 2. You can choose to provide a more advanced type of filter expression using predefined operators.

# **Viewing Excel DataSource Data**

When an Adapter Binding is established that references an Excel table (Worksheet), a hyperlink with the name of the Excel sheet will be displayed in the column labeled "*Data View*". Left Mouse-click this link to view all data contained in this sheet in the Data Source Grid View. This DataView will be filtered based on the provided filter expression.

## **Filtering the attached Excel Data Source**

You can optionally provide filter criteria on the attached Excel file data source that defines the scope of entities available while processing the conversion task. The filter can be applied in one of two ways as follows:

- 1. Provide the Key column and Key value as the source of the filter
  - Select the data source column in the [Template Key] column as the source of the filter value.
  - Specify the value to filter the data source in the [Template Key Value] column based on values found in the specified [Template Key] column.
- 1. Provide a filter expression directly in the [Template Key Value] in the form FILTER=Boxnum='28'
  - The keyword FILTER= must precede the filter criteria.
  - The excel value to filter on must be enclosed within single quote characters as in '28'
  - $\bullet$  See  $\underline{\text{examples}}$  of filter expressions supported

### **Tips and Tricks**

• When viewing the data for your Excel DataSource table, the maximum number of visible data columns is limited to 200 columns by default. You can change this value to expose more columns in the DataView. This maximum limit optimizes the performance of the

DataView when large Data Tables are being viewed. Clicking the Max Visible Columns: 200 to button will allow you to choose the columns to display in the DataView.

- You can specify a hierarchical Control Module path by using the '\' character to separate Control Module Container names as in "HEATERS\HEATER1\FIRING CONTROL"
- You can filter your DataView to eliminate values that are empty using the Filter Expression: LEN (TRIM (PARENTASSET)) > 0 where PARENTASSET is a column in your Data Table.
- When testing your Control Conversion solution you may want to reduce your test results to a minimum by selecting one or more entities from your Data Table for processing. You can do this by selecting the entities in your DataView and clicking the button. This will define the Entity scope of the Adapter Binding such that only these entities will be processed when generating the Data Collection. You can view

the Entity scope by clicking the drop-down list Selection . To change the Entity scope you can either select a new group of entities from the DataView or select Entity(s) from within the drop-down list and press the Delete button to remove them.

- If your selected Primary Key column fails on an error that indicates that the column values are not unique, then it is possible that your column contains row data with space characters. Try selecting all rows that contain no valid data in the Excel Worksheet and use the Excel context menu (right mouse-click) function **Delete** to assure that empty rows are deleted from the Worksheet.
- If an error of type "FailedToFindPrimaryKeyBinding" occurs when generating a Data Collection, verify that you have defined a Control Binding with the <u>Primary Key attribute</u> in your Control Binding Template that is being referenced in the Control Conversion Adapter.
- The Primary Key column name that you choose for an Adapter Binding should be the DataSource column name that is bound to the Target Control Property in your Control Binding Template that uniquely defines the Control Strategy name. Failure to do this will result in the Data Collection generation failing to identify new or existing Control strategies in your output path.

## Notes

The Suffix Option is currently not supported in the Control Conversion Template.

# **Excel Data Source filter expressions**

# **Examples of filter expressions**

This example describes syntax of Filter expression. It shows how to correctly build expression string using methods to escape values.

#### Column names

If a column name contains any of these special characters  $\sim$  ( ) #\/ = > < + - \* % & | ^ ' " [ ], you must enclose the column name within square brackets [ ]. If a column name contains right bracket ] or backslash \, escape it with backslash (\] or \\).

```
Filter = "id = 10"; // no special character in column name "id" Filter = "$id = 10"; // no special character in column name "$id" Filter = "[#id] = 10"; // special character "#" in column name "#id" Filter = "[[id]] = 10"; // special characters in column name "[id]"
```

#### Literals

String values are enclosed within single quotes ''. If the string contains single quote ', the quote must be doubled.

```
Filter = "Name = 'John'" // string value
Filter = "Name = 'John "A"" // string with single quotes "John 'A""
```

Date values are enclosed within sharp characters ##.

```
Filter = "Date = #12/31/2008#" // date value (time is 00:00:00)

Filter = "Date = #2008-12-31#" // also this format is supported

Filter = "Date = #12/31/2008 16:44:58#" // date and time value
```

Alternatively you can enclose all values within single quotes ''. It means you can use string values for numbers or date time values. In this case the current culture is used to convert the string to the specific value.

```
Filter = "Date = '12/31/2008 16:44:58"" // if current culture is English
Filter = "Date = '31.12.2008 16:44:58"" // if current culture is German

Filter = "PVEUHI = '1199.90" // if current culture is English
Filter = "PVEUHI = '1199.90" // if current culture is German
```

### **Comparison operators**

Equal, not equal, less, greater operators are used to include only values that suit to a comparison expression. You can use these operators = <> < <= >>=.

Note: String comparison is culture-sensitive.

```
Filter = "BOXNUM = '10"  // number is equal to 10

Filter = "Date < #1/1/2008#"  // date is less than 1/1/2008

Filter = "Name = '2FC100"  // string is equal to '2FC100'

Filter = "Name >= '2F"  // string comparison
```

Operator IN is used to include only values from the list. You can use the operator for all data types, such as numbers or strings.

```
Filter = "BOXNUM IN (10, 12, 14)"  // integer values

Filter = "PVEULO IN (0.0, -10.0, -100.0)"  // float values

Filter = "Name IN ('FC100', 'FC110', 'FC120')"  // string values

Filter = "Date IN (#12/31/2008#, #1/1/2009#)"  // date time values

Filter = "BOXNUM NOT IN (21, 22, 23)"  // values not from the list
```

Operator LIKE is used to include only values that match a pattern with wildcards. Wildcard character is \* or %, it can be at the beginning of a pattern 'value', at the end 'value', or at both 'value'. Wildcard in the middle of a pattern 'va\*lue' is not allowed.

```
Filter = "Name LIKE '2F*" // values that start with '2F'
Filter = "Name LIKE '%FC%" // values that contain 'FC'

Filter = "Name NOT LIKE '2*" // values that don't start with '2'
```

If a pattern in a LIKE clause contains any of these special characters \* % [], those characters must be escaped in brackets [] like this [\*], [%], [[] or []].

Copyright © 2024 by Automation Software Technology, Inc. All Rights Reserved.

```
Filter = "Name LIKE '[*]*" // values that starts with '*'
Filter = "Name LIKE '[[]*" // values that starts with '['
```

#### **Boolean operators**

Boolean operators AND, OR and NOT are used to concatenate expressions. Operator NOT has precedence over AND operator and it has precedence over OR operator.

```
Operator AND has precedence over OR operator, parenthesis are needed Filter = "HWYNUM = '02' AND (BOXNUM = '20' OR BOXNUM = '24')";

Following examples do the same 
Filter = "HWYNUM <> '02' AND PNTBOXTY = 'EC'";

Filter = "HWYNUM = '02' AND NOT PNTBOXTY = 'EC'";

Filter = "NOT (ALGIDDAC = 'PIDNORM' OR ALGIDDAC = 'DAS')";

Filter = "ALGIDDAC NOT IN ('PIDNORM', 'DAS')";
```

#### Arithmetic and string operators

Arithmetic operators are addition +, subtraction -, multiplication \*, division / and modulus %.

```
Filter = "PVEUHI - PVEULO < 10"; // PV range < 10.0
Filter = "PVEUHI % 100 = 0"; // PV range evenly divisible by 100
```

#### **Aggregate Functions**

These aggregate functions are supported: SUM, COUNT, MIN, MAX, AVG (average), STDEV (statistical standard deviation) and VAR (statistical variance).

This example shows aggregate function performed on a single table.

```
Select entities with above-average PV range

Filter = "PVEUHI - PVEULO > AVG(PVEUHI - PVEULO)";

Select entities which have more than 1 input

Filter = "COUNT(NMBRINPT) > 1";
```

## **Functions**

There are also the following functions that can be applied to a data column.

```
CONVERT – converts particular expression to a specified .NET Framework type
      Syntax: CONVERT(expression, type)
      Example: CONVERT(NUMINPTS, 'System.Int32')
LEN – gets the length of a string
      Syntax: LEN(expression)
      Example: LEN(TAG) > 0
ISNULL - checks an expression and either returns the checked expression or a replacement value
      Syntax: ISNULL(expression, replacementvalue)
      Example: ISNULL(PVEUHI, 0)
IIF – gets one of two values depending on the result of a logical expression
      Syntax: IIF(expr, truepart, falsepart)
      Example: IIF(PVEUHI>1000, 'D0', 'D1')
TRIM - removes all leading and trailing blank characters like \r, \n, \t, , '
      Syntax: TRIM(expression)
      Example: TRIM(DESC) = 'ONLINE'
SUBSTRING - gets a sub-string of a specified length, starting at a specified point in the string
      Syntax: SUBSTRING(expression, start, length)
      Example: SUBSTRING(NAME, 1, 2)
```

## **Tips & Tricks**

#### **Common Failures**

- The column in the Excel sheet cannot automatically infer its type. When columns contain a mix of data such as a numeric and string, Excel may not be able to establish the column type as a Text type. Functions that work with string values may fail, which requires you to select the Excel column and set its format type to *Text*.
- When working with number types such as data converted from a Honeywell Hiway EB file, data values such as HWYNUM, BOXNUM, etc. may need to be enclosed within single quote characters as in HWYNUM = '06' and BOXNUM = '22' when used in a Filter expression.



#### **Data Collections**

Data Collections are generated from within the Control Conversion Template view using configured adapter bindings that define the relationships between an Excel Data Source file and its data tables and a Control Binding Template file that specifies how Data Source column data is mapped to PKS based configuration properties.

## **Generating a Data Collection**

To execute a Data Collection task click the button in the Control Conversion Template view. The Data Collection generation task will use the data configured in the Control Conversion Template to analyze and display Control strategies and their Parameter data that has been detected as changed or new.

### Working with the Data Collection View

The Data Collection view contains the scope of entities and their parameters that are used in a Workflow Control Conversion task for bulk Build/Edit type operations.

The Data Collection view organizes its collected data into *New* and *Existing* categories in the TreeView located in the left pane view. You can view the parameter data in the ListView located in the right pane view by selecting one or more entities as show below. Parameter items that have their *Enabled* state checked are chosen for processing during a Control Conversion task execution. You can choose to enable/disable both entities and Parameter items by checking/unchecking the checkboxes for the Entity item and/or its Parameter items.

When a Data Collection view is used in a Control Conversion task any Parameter items that were enabled for processing will be time stamped to indicate the status of its conversion and the time it occurred as indicated in the bottom example image view.

#### **Options**

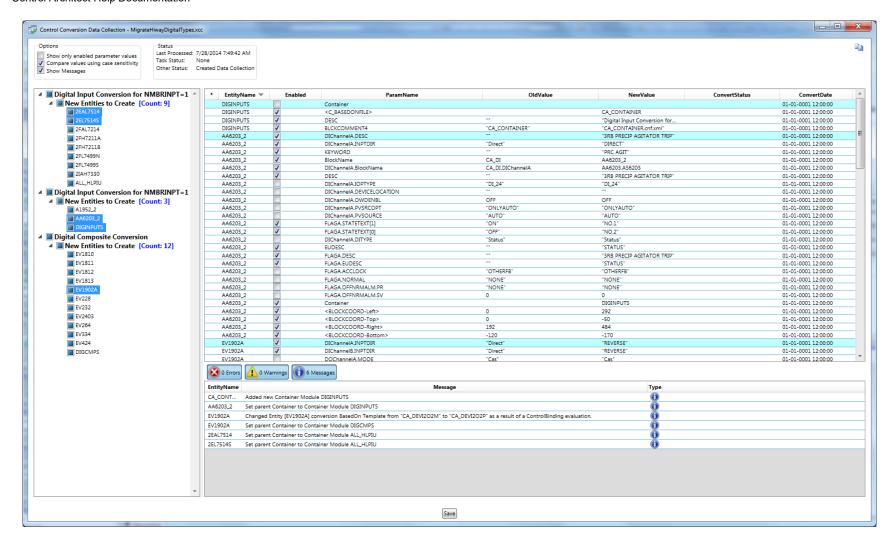
- 1. Show only enabled parameter values Use this option to filter the Parameter ListView such that only items that have their *Enabled* state checked will be displayed.
- 2. Compare values using case sensitivity Use this option to determine how Parameter value case sensitivity is used when comparing string type values.
- 3. Show Messages = Use this option to display messages associated with the entities that were generated during the Data Collection .

#### **Status**

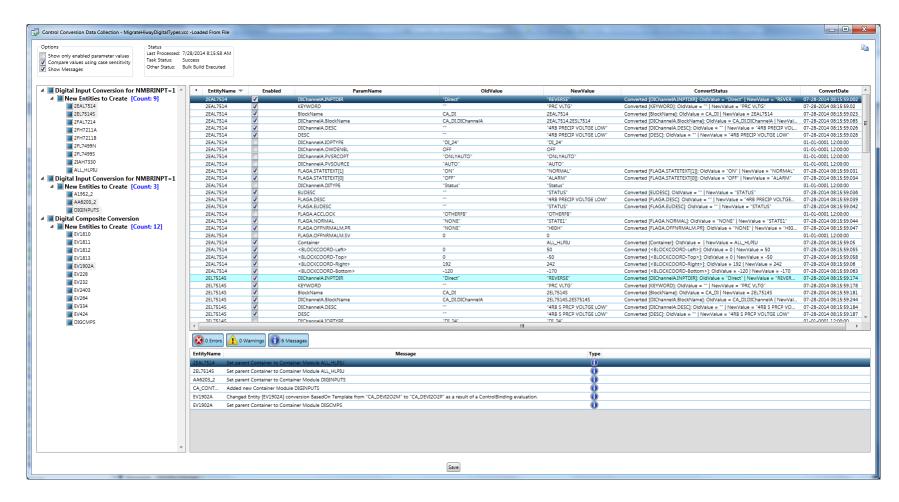
The status of a Data Collection indicates when it was used during the execution of a Workflow Control Conversion task.

#### **Examples**

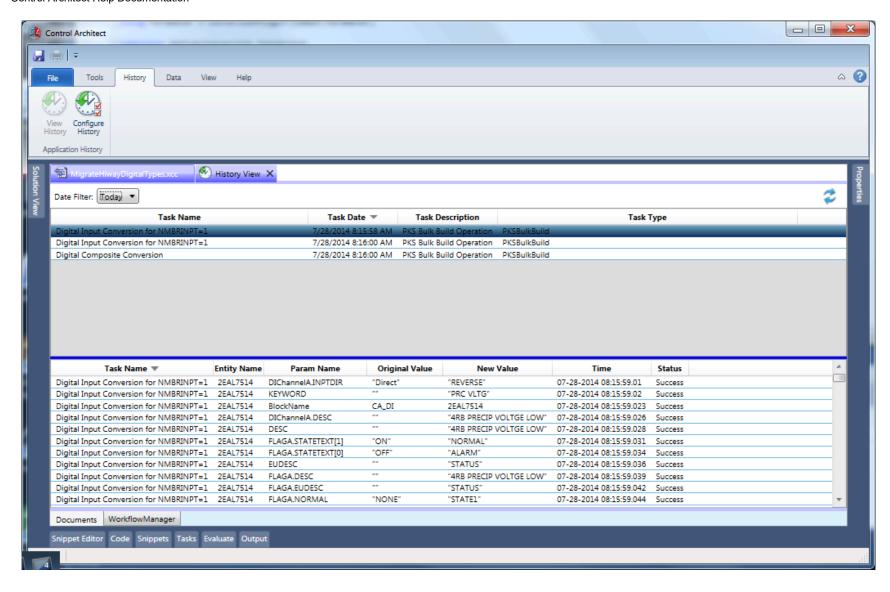
Below are examples of a Data Collection that has been generated from a set of Control Conversion Template Adapter bindings showing it preexecution status, the status after the Data Collection has been executed in a Workflow Manager Conversion Activity, and the History collected for the task execution.



This Data Collection was used in a Control Conversion task where its Parameter items that were enabled for processing were time stamped to indicate the status of its conversion and the time it occurred



All Bulk Build/Edit tasks that have been executed through the Workflow Manager maintain History for all Entity related parameter changes.



# Parsing EB files to Excel

Honeywell Exception Build (EB) files can be converted to Microsoft Excel Workbook files using a <u>Control Conversion Workflow activity</u>. This activity allows you to specify multiple EB files to be processed during the same task execution. Optionally, a specific point type or Template name can be specified to define what type is read and converted to Excel.

When EB files are converted, the generated Excel Workbook file will contain a tab or Spreadsheet for each point type encountered in the EB file.

## **Data Source Control Graph View**

#### **Overview**

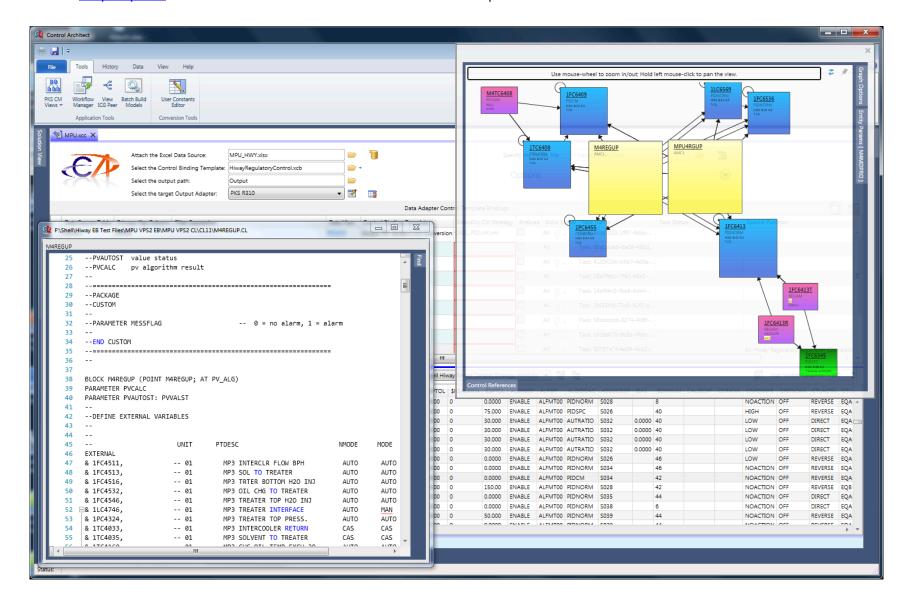
The Control Graph provides a visual rendering of entities found in Hiway Box configuration that contain parameter connection relationships that form a Control Strategy. The Control Graph view analyzes a selected Hiway Box to form a visual block rendering of its contained Slot configuration with their parameter connection relationships rendered as data flow connections.

The Control Graph view provides a summary view of Box Slot entities that have connection relationships that form a Control Strategy. It provides a 10,000 foot view of your control system applications allowing you to quickly visualize relationships.

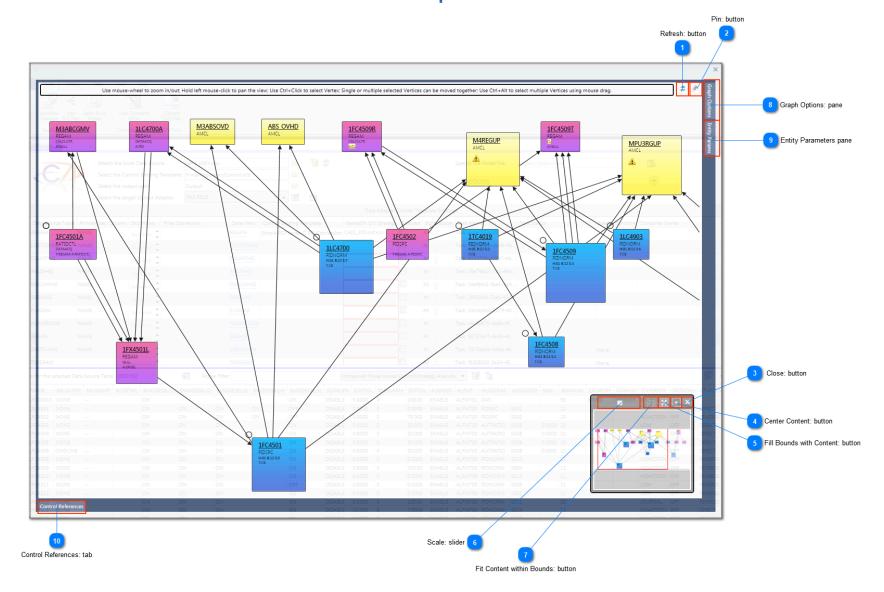
The only Analyzer supported in this release is for Hiway analysis. The Hiway Analyzer supports not only relationships between Hiway Box slots, but also relationships that exist within AM entities, AMCL code and Hiway PIU type references. The visual blocks are rendered to quickly identify the node type based on its block surface information and color schema. AMCL type references support an Adorner that becomes visible when the block is hovered over with the mouse cursor. This Adorner displays a button that allows you to view both AMCL source code in a Code View window and also provides a list of CL packages that are referenced in AM entities that can be selected to view its Package source code.

See how to work with a <u>Vertex Block</u> in the Control Graph.

See the **Graph Options** for how to control the visual content of the Control Graph.



# **Control Graph Overview**



# Refresh: button



Click this button to redraw the Control Graph surface based on the original strategy analysis. After viewing deeper relationships using the expansion functions and/or removing Entity Vertex nodes, you can view the original analysis contents by clicking this Refresh button.

### Pin: button



Click this button to pin the Control Graph surface contents and prevent any automatic updates when selecting new Data Source or Entity items that is the source of the Control Graph analysis. To update the Control Graph with new analysis results, click this button again to unpin the Control Graph surface.

## Close: button



Click this button to close the Navigation View. You can later restore this view.

# Center Content: button



Click this button to center the Control Graph contents.

## Fill Bounds with Content: button



Click this button to zoom the Control Graph contents to fit the current view.

# Scale: slider



Use this slider to change the current zoom scale for the Control Graph surface.

# Fit Content within Bounds: button



**Currently Not Used** 

Graph Options: pane



Click this Tab to view options for the Control Graph view.

Entity Parameters pane



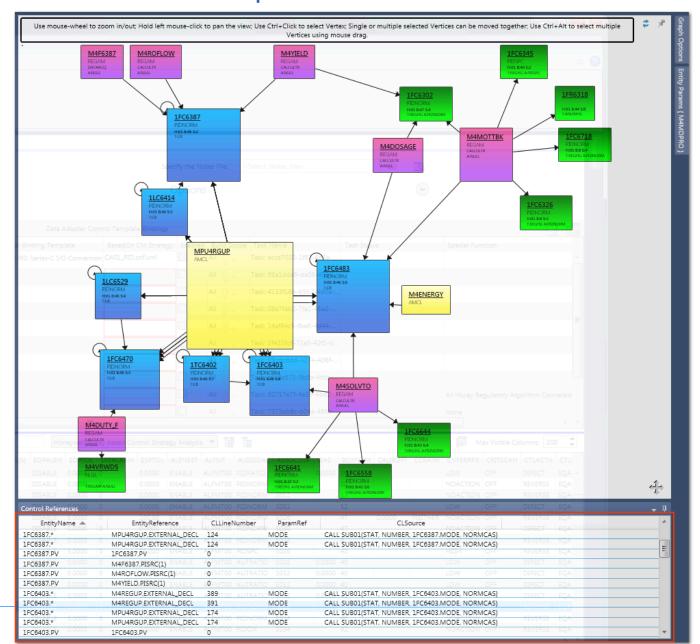
Click this Tab to view the Entity Parameters view. Items in this list can be copied/pasted through Windows Clipboard using the keyboard sequence **Ctrl+C/Ctrl+V**.

**Control References: tab** 

Control References

Click this tab to view a list of connection relationships currently displayed on the Control Graph surface. Selecting an item in the list will bring into focus and flash the Entity Vertex Control that is identified as the Entity Name. Items in this list can be copied/pasted through Windows Clipboard using the keyboard sequence **Ctrl+C/Ctrl+V**.

# **Control Graph References**



Control References: list

Control References: list

EntityName 📥	EntityReference	CLLineNumber	ParamRef	CLSource	NOACTIO			
1FC6387.*	MPU4RGUP.EXTERNAL_DECL	124	MODE	CALL SUB01(STAT, NUMBER, 1FC	6387.MODE, NO	RMCAS)		EQA
1FC6387.*	MPU4RGUP.EXTERNAL_DECL	124 PIDNOR	MODE	CALL SUB01(STAT, NUMBER, 1FC	6387.MODE, NO	RMCAS)		
1FC6387.PV 5.0000	0 1FC6387.PV 000 ENABLE	AOFMTOO PIDSPC	5026	40	HIGH	OFF	REVERSE	EQA
1FC6387.PV 0.0000	M4F6387.PISRC(1) ENABLE	AgFMT00 AUTRATI	IO   S032	0.0000   40	LOW	OFF	DIRECT	EQA
1FC6387.PV 0.0000	<ul> <li>M4ROFLOW.PISRC(1)   ABLE</li> </ul>	AOFMTOO AUTRATI						EQA
1FC6387.PV 0.0000	0 M4YIELD.PISRC(1) ENABLE	AQFMT00 AUTRATI	IO S032	0.0000 40	LOW	OFF	DIRECT	EQA
1FC6403.* 0.0000	M4REGUP.EXTERNAL_DECL	A389ITOO AUTRAT	MODE	CALL SUB01(STAT, NUMBER, 1FC	6403.MODE, NO	RMCAS)	DIRECT	FOA
1FC6403.*	M4REGUP.EXTERNAL_DECL	391	MODE	CALL SUB01(STAT, NUMBER, 1FC	6403.MODE, NO	RMCAS)	DEV/EDGE	FOA
1FC6403.*	MPU4RGUP.EXTERNAL_DECL	174	MODE	CALL SUB01(STAT, NUMBER, 1FC	6403.MODE, NO	RMCAS)		
1FC6403.*	MPU4RGUP.EXTERNAL_DECL	174 PIDNOR	MODE	CALL SUB01(STAT, NUMBER, 1FC	6403.MODE, NO	RMCAS)	REVERSE	EQA
1FC6403.PV	1FC6403.PV ENABLE	A <sub>0</sub> FMT00   PIDCM	5034	42	NOACTI	ON OFF	REVERSE	EQA

Displays a list of all Control connection references displayed on the Control Graph surface. Use Ctrl+C/Ctrl+V to copy/paste using the Windows

# **Embedding Modules**

# **Hierarchical Building Overview**

Hierarchical building allows the creation of a Hierarchical View of a process from top down, or bottom up. The Hierarchy may reflect the Plant Data Model and/or Configuration Asset Model.

The Containment View, in Control Builder's tree view, shows the containment relationship among all configured Control Modules including CM's, SCM's.

Parameter Projection allows the projection of parameters of an embedded block upward onto its container. When this container CM/SCM is embedded in another CM container, the projected parameters can then be exposed as pins on the block surface of this embedded block in the new container, and connections can be graphically made to these pins. Projected Parameters are defined as if they are the inherent parameters and will have all the attributes of the origin parameters, but will be referenced by their new names. The actual connections are made to the inherent origin parameters.

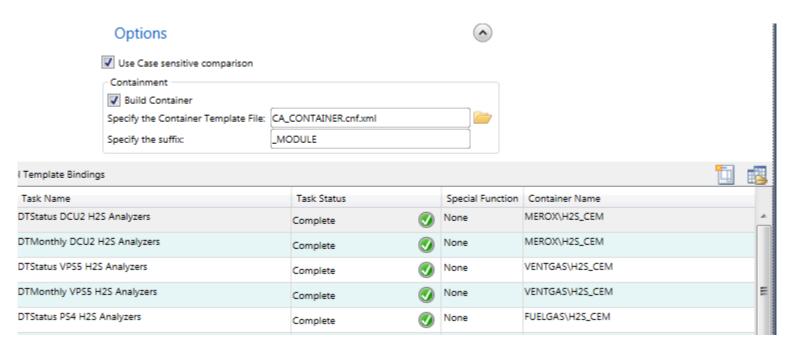
Embedded Control Module development promotes good Engineering practice and forces the engineer to make good Control Strategy development decisions such as module cohesiveness. It defines the relationships amongst Control Modules (Control Strategies) that helps to understand their control relationships compared to looking at hundreds or thousands of Control Module strategies in a flat list. Imagine if Windows File Explorer did not support the function of Directory or Folder paths that define file relationships.

#### **Control Architect Embedded Development Support**

Control Architect supports Embedded (hierarchical) Control Module development through various options specified in a Control Conversion Template. You can choose to specify a Control Module path (hierarchy) by simply specifying the Module path as in "HEATERS\HEATER1" in the Control Conversion Adapter Binding or you can specify it as an Excel DataSource column value that can be bound using a Control Binding that will set the Control Module *Container* value.

#### Specify the Control Module hierarchy in a Control Conversion Adapter Binding

The example below shows how a Control Module path is used to create and set the containment relationships for Control Conversion tasks.



When a path is specified in the "Container Name" column, if the path Control modules do not exist, they will be defined as new Control modules to create in the generated Data Collection. When the Data Collection is executed it will create these new Control Modules and embed them as defined in the path relationship. You must choose the **Build Container** option and specify the Control Module configuration template file to use when building the Control modules. Control Architect provides a default set of Template files that can be used when generating Honeywell PKS CEE configuration types.

#### Specify the Control Module hierarchy in a Control Binding

You can add a Control Property name Container to the set of Control properties for a PKS CEE type Template that can be bound to a DataSource value in a Control Binding. This assumes that the *Container* value (Control Module) being set already exists. This type of containment building does not support path building as in "HEATERS\HEATER100". If you want to specify the containing Control Module you should specify "HEATER100" as the containing Control Module value. The Control Module specified as the container must already exist or you can create another Conversion task that simply builds the containing Control Modules.

The example below shows how a Control Module name is specified in an Excel DataSource and used as the source property of a Control Binding bound to a Control Module *Container* target Property. The Excel DataSource column named "CONTAINEDIN" specifies the containing Control Module name that is bound to the PKS CEE Control Module Parameter "Container".

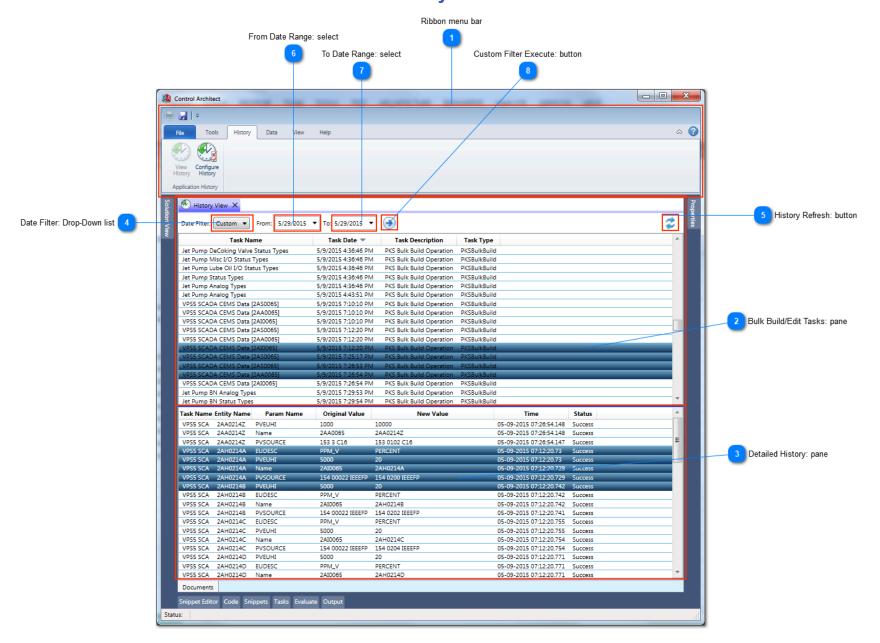
4 8	SourceProperty	TargetProperty	DefaultValueIfEmpty	Code	Enabled	EnableRounding	RoundToPlaces
	DESC	DESC		// Binded to Source Control Property	J		2
	PARENTASSET	PARENTASSET		// Binded to Source Control Property	J		2
	DESC	4XDRIFT.DESC		// Binded to Source Control Property	J		2
	DESC	FLAGA.DESC		// Binded to Source Control Property	<b>J</b>		2
F	NAME	BlockName		// Binded to Source Control Property	J		2
	CEE	AssignedTo		// Binded to Source Control Property	J		2
	OPCSTREAMSTATUS	LTA. <pref-il-lta.in[1]></pref-il-lta.in[1]>		// Binded to Source Control Property	J		2
	OPCSTREAMSTATUS	CHECKBADA. <pref-il-checkbada.in></pref-il-checkbada.in>		// Binded to Source Control Property	J		2
	DRIFT-2X	ORA. <pref-il-ora.in[2]></pref-il-ora.in[2]>		// Binded to Source Control Property	J		2
	DRIFT-4X	ORA. <pref-il-ora.in[3]></pref-il-ora.in[3]>		// Binded to Source Control Property	<b>J</b>		2
	IN_CAL	RTRIGA. <pref-il-rtriga.in></pref-il-rtriga.in>		// Binded to Source Control Property	J		2
	CONTAINEDIN	Container		// Binded to Source Control Property	<b>J</b>		2
	DRIFT-4X	4XDRIFT. <pref-il-4xdrift.pvfl></pref-il-4xdrift.pvfl>		// Binded to Source Control Property	1		2

# **History**

Control Architect supports History collection for all Bulk Build/Edit activities.

Collected History data can be filtered and viewed from the History View while old History can be deleted using the History Configuration View.

# **History View**

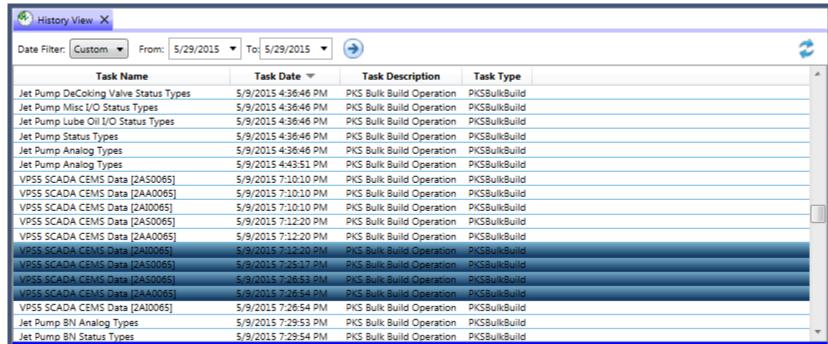


### 🦱 Ribbon menu bar



Select the "History" tab from the main Ribbon Toolbar to view and configure History.

## Bulk Build/Edit Tasks: pane



This Pane contains all Bulk Build/Edit tasks that have been executed during the selected data period. Select one or more task items in this list to view its detailed Parameter history.

Items in this Pane can be copied to the Window Clipboard and pasted into Excel using the keyboard commands Ctrl+C / Ctrl+V.

# Detailed History: pane

Task Name	Entity Name	Param Name	Original Value	New Value	Time	Status	
VPS5 SCA	2AA0214Z	PVEUHI	1000	10000	05-09-2015 07:26:54.148	Success	
VPS5 SCA	2AA0214Z	Name	2AA0065	2AA0214Z	05-09-2015 07:26:54.148	Success	
VPS5 SCA	2AA0214Z	PVSOURCE	153 3 C16	153 0102 C16	05-09-2015 07:26:54.147	Success	
VPS5 SCA	2AH0214A	EUDESC	PPM_V	PERCENT	05-09-2015 07:12:20.73	Success	
VPS5 SCA	2AH0214A	PVEUHI	5000	20	05-09-2015 07:12:20.73	Success	
VPS5 SCA	2AH0214A	Name	2AI0065	2AH0214A	05-09-2015 07:12:20.729	Success	
VPS5 SCA	2AH0214A	PVSOURCE	154 00022 IEEEFP	154 0200 IEEEFP	05-09-2015 07:12:20.729	Success	
VPS5 SCA	2AH0214B	PVEUHI	5000	20	05-09-2015 07:12:20.742	Success	
VPS5 SCA	2AH0214B	EUDESC	PPM_V	PERCENT	05-09-2015 07:12:20.742	Success	
VPS5 SCA	2AH0214B	Name	2AI0065	2AH0214B	05-09-2015 07:12:20.742	Success	
VPS5 SCA	2AH0214B	PVSOURCE	154 00022 IEEEFP	154 0202 IEEEFP	05-09-2015 07:12:20.741	Success	
VPS5 SCA	2AH0214C	EUDESC	PPM_V	PERCENT	05-09-2015 07:12:20.755	Success	
VPS5 SCA	2AH0214C	PVEUHI	5000	20	05-09-2015 07:12:20.755	Success	
VPS5 SCA	2AH0214C	Name	2AI0065	2AH0214C	05-09-2015 07:12:20.754	Success	
VPS5 SCA	2AH0214C	PVSOURCE	154 00022 IEEEFP	154 0204 IEEEFP	05-09-2015 07:12:20.754	Success	
VPS5 SCA	2AH0214D	PVEUHI	5000	20	05-09-2015 07:12:20.771	Success	
VPS5 SCA	2AH0214D	EUDESC	PPM_V	PERCENT	05-09-2015 07:12:20.771	Success	
VPS5 SCA	2AH0214D	Name	2AI0065	2AH0214D	05-09-2015 07:12:20.771	Success	

This Pane contains the detailed Parameter history for the selected Bulk Build/Edit tasks.

Items in this Pane can be copied to the Window Clipboard and pasted into Excel using the keyboard commands Ctrl+C / Ctrl+V.

# Date Filter: Drop-Down list



Select from the drop-down list the date range to filter the tasks that contain History to view. Select the "Custom" item in the drop-down list to specify a custom Date range to filter on. When "Custom" is chosen, a from and to date edit box becomes visible that allows you to specify your custom Date range.

## History Refresh: button



Click this button to refresh History. You would normally use this button to refresh and view History for a Bulk Build/Edit task that was executed while this view was visible.

# From Date Range: select



Click the drop-down arrow to select the From Date range.

## To Date Range: select



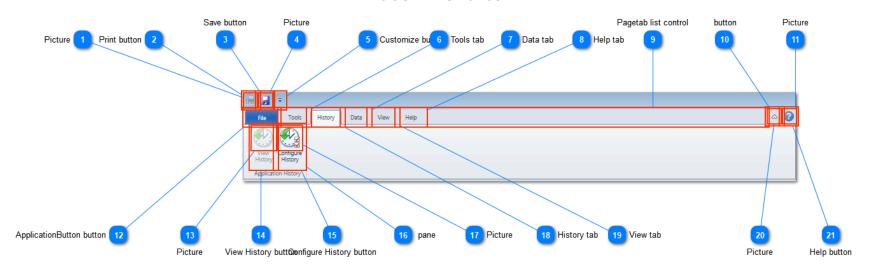
Click the drop-down arrow to select the To Date range.

## Custom Filter Execute: button



Click this button to filter the History view using your custom selected Date range.

### Ribbon menu bar



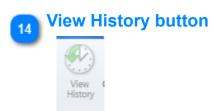
- Picture
- Print button
- Save button
- 4 Picture
- Customize button
- Tools tab
- 7 Data tab
- 8 Help tab
- Pagetab list control

  Tools History Data View Help
- button





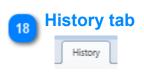


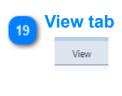








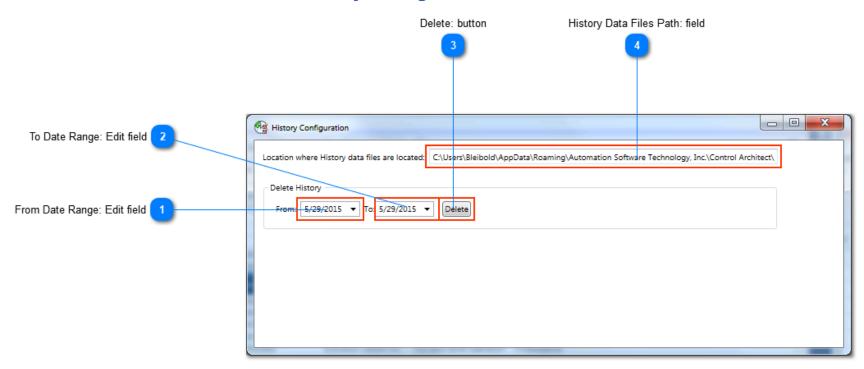








# **History Configuration window**



From Date Range: Edit field



Specify the From Date part of the Date range to use for the Delete filter.

To Date Range: Edit field



Specify the To Date part of the Date range to use for the Delete filter.

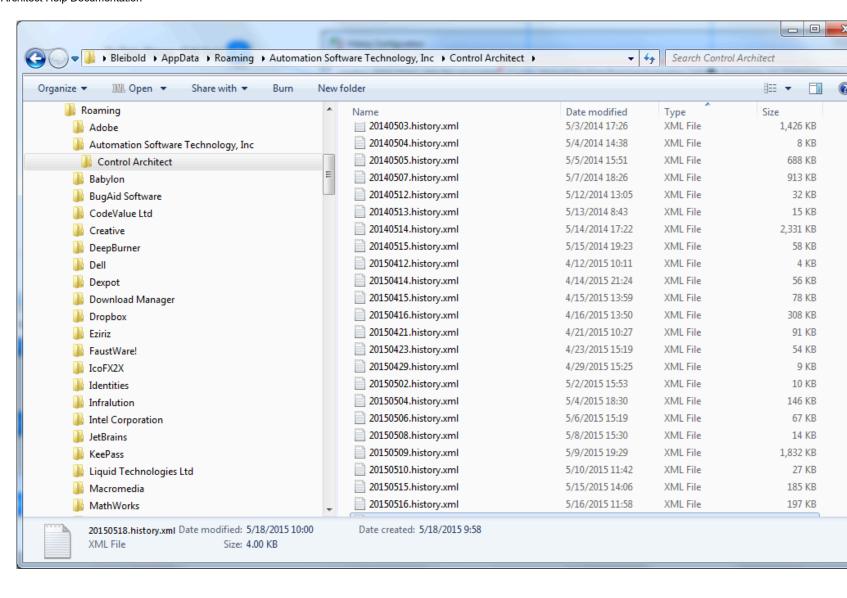
Delete: button



Click this button to permanently delete History data files for the selected Date range. Once deleted, these data files cannot be recovered.

History Data Files Path: field

This data field shows the folder path on your file system where History data files are created and stored. You can navigate to this path to archive collected on a daily basis where a single History data file will contain the History for a 24 hour period. The image below shows how History where the calendar Date is used to form the data file name:

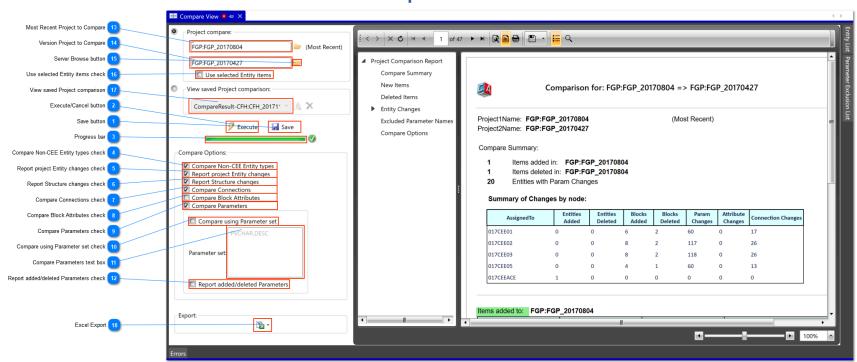


# **Compare View Overview**

The Compare View provides a set of functions that allow users to compare Projects or snapshots to locate changes made between 2 versions of configuration data. This differential report provides an effective tool for managing changes during the full life cycle of a Control system. Only configuration data is supported for data comparison including PKS CEE configuration, PKS SCADA configuration, and LCN Entity configuration.

Comparison results are generated to a report that can be saved and exported to media such as Microsoft Word, Adobe PDF, etc.

## **Compare View**



# Save button



Click this button to save the comparison result for later recall.

Comparison results are saved to the document Server and can quickly be loaded to the Report View by selected the result set from the Drop-down list and clicking the adjacent Preview button.

Saved comparison results can be deleted by clicking the Delete button located adjacent to the dropdown list. This will delete the currently selected item in the dropdown list.

Multiple saved results can be differentiated by viewing the ToolTip for an item in the dropdown list. The ToolTip will indicate the time stamp when saved.



# Execute/Cancel button



Click this button to start the Compare execution and click to cancel the execution of an active Comparison task.

# Progress bar

Indicates the current progress of an active Comparison task.

# Compare Non-CEE Entity types check

# Compare Non-CEE Entity types

Choose this option to compare for Non-CEE type entities. Examples of Non-CEE entities would be SCADA and LCN configuration types.

# Report project Entity changes check

# Report project Entity changes

Choose this option to check for entities that have been added or deleted between project versions.

# Report Structure changes check

# Report Structure changes

Choose this option to check for Blocks that have been added or deleted from a Control Module.

Compare Connections check

Compare Connections

Choose this option to compare for Connection value changes between CM's and blocks. If a connection reference has changed, then using this option will identify and report the change.

Compare Block Attributes check

Compare Block Attributes

Choose this option to compare for attribute changes on a Block such as Block surface color, exposed Parameter changes, etc.

Compare Parameters check

Compare Parameters

Choose this option to compare for Parameter value changes.

Compare using Parameter set check

Compare using Parameter set

Choose this option to specify a set of comma delimited Parameter names to compare values for.

This option ignores Parameter items that are defined in the Parameter Exclusion definitions.

Compare Parameters text box



Specify a comma delimited set of Parameter names to compare values for.

Report added/deleted Parameters check

Report added/deleted Parameters

Choose this option to report Parameters that have been added or deleted.



Using this option when comparing between major PKS releases may result in large amounts of parameter data in the report. Examples would be when comparing a Server snapshot for PKS R310 against a snapshot for PKS R410 where R410 added a significant number of Alarming related parameters to various Function Block types that would be reported as "Added" parameters.

Most Recent Project to Compare

FGP:FGP\_20170804

Click the Server browse button to select the most recent project snapshot to compare.

Version Project to Compare

FGP:FGP\_20170427

Click the Server browse button to select the version project snapshot to compare against the most recent snapshot.

Server Browse button

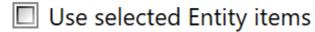


Click this button to select a Project Collection (snapshot) from the document Server.

Copyright © 2024 by Automation Software Technology, Inc. All Rights Reserved.



# **Use selected Entity items check**



Choose this item to compare using the items in the Entity list that are enabled (checked).

17

# **View saved Project comparison**



Click this drop-down list to view available saved Project comparison reports that have been previously generated and saved.

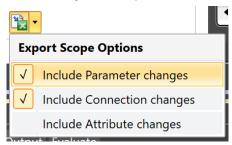


## **Excel Export**



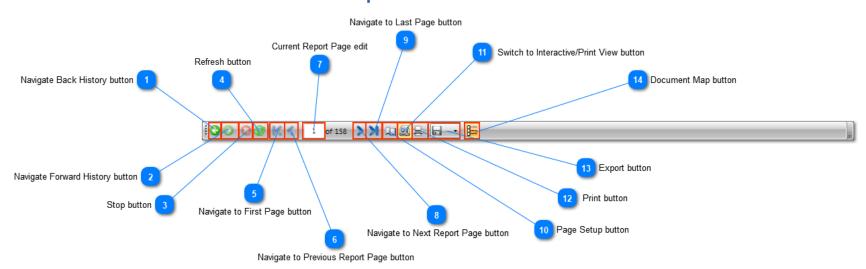
Click this button to export the compare results data to an Excel Workbook file.

Click the adjacent dropdown button to select what compare data is generated to the Excel Workbook.



Use this export function to be able to filter and query data in Excel.

# **Report View ToolBar**







Click this button to navigate back in history.

Navigate Forward History button



Click this button to navigate forward in history.

Stop button



Click this button to stop the report generation.

Refresh button



Click this button to refresh the report if the source data has changed. This is not applicable to this report type.

Navigate to First Page button



Click this button to navigate to the first page of the report.

Navigate to Previous Report Page button



Click this button to navigate to the previous page in the report.

Current Report Page edit



Enter the page number to navigate to a specific report page.

Navigate to Next Report Page button



Click this button to navigate to the next page in the report.

Navigate to Last Page button



Click this button to navigate to the last page of the report.

Page Setup button



Click this button to set the page options for the report.



# **Switch to Interactive/Print View button**



Click this button to toggle between interactive and print preview mode.

12

## **Print button**



Click this button to print the report.

13

## **Export button**



Click this button to export the report to other media types such as Microsoft Word, Adobe PDF, etc.

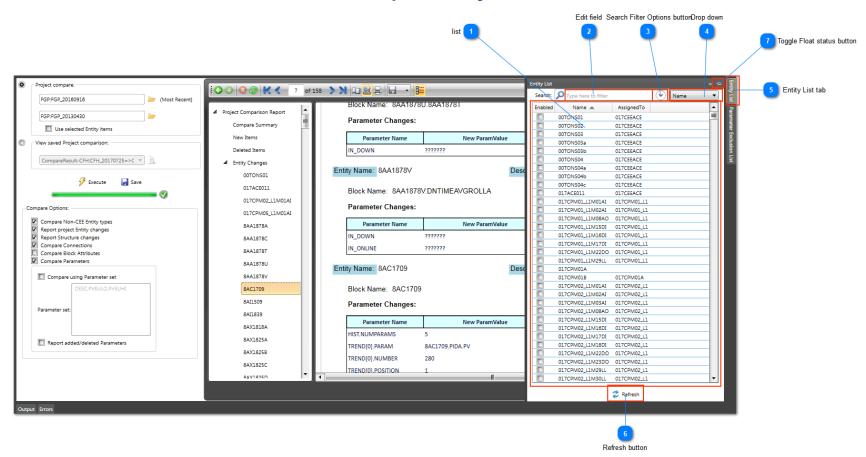
14

# **Document Map button**

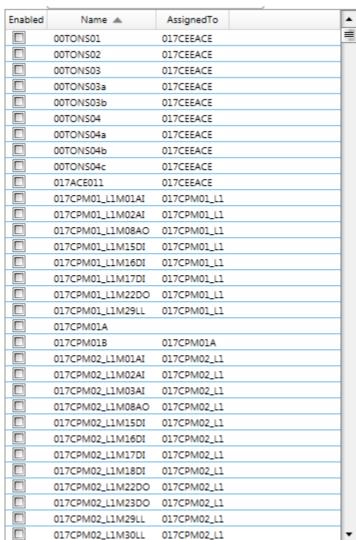


Click this button to toggle the Document Map visibility.

## **Compare Entity List**



list



The list of entities that were found in the selected Most Recent Project/Collection.

You can choose to compare a subset of entities by selecting the items and clicking the space bar key to toggle the checked status and enable the items for comparison. Check the option *Use selected Entity items* prior to execution.

2 Edit field

Type here to filter

Enter the text to filter on.

Search Filter Options button

Copyright © 2024 by Automation Software Technology, Inc. All Rights Reserved.



Click this drop-down button to view options for specifying search/filter text.

4 Drop down

Click this drop-down to select a data column to filter on. The text that is entered in the Edit field will be applied to the data column that is selected here.

Entity List tab



Click this tab to show the Entity List View.

Refresh button



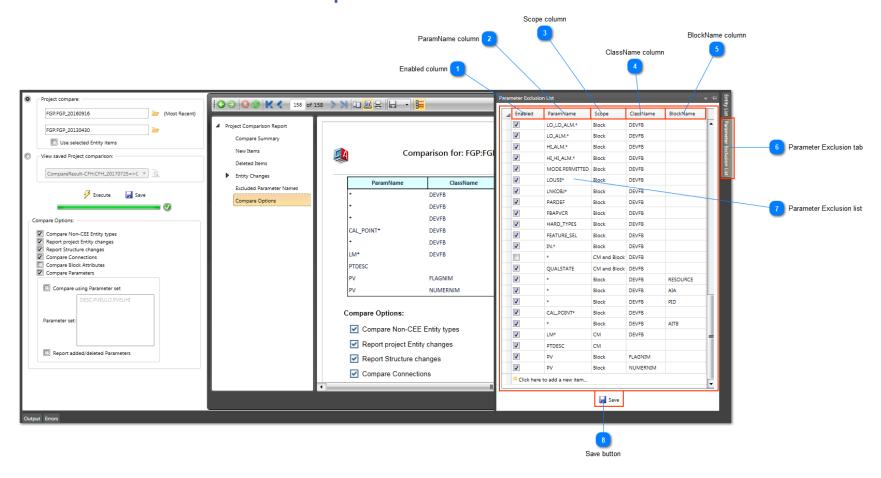
Click this button to refresh the list of entities and remove all checked enabled states.

Toggle Float status button



Click this pin button to toggle the Entity List view floating and docked state.

## **Compare Excluded Parameter List**





#### **Enabled column**



Check this Parameter item to enable its exclusion status. If a Parameter Exclusion item is disabled, then it will be ignored when looking for Parameter items to exclude during comparison. You twill typically use this to toggle on/off the exclusion of a particular Parameter item based on the type of configuration data that is being compared.



#### ParamName column



Specify the Parameter name that an exclusion will apply to.

Parameter names can be specified using the wild card character '\*' to allow for specifying that any value or partial values are to be matched when locating Parameter names that might apply to a Parameter Exclusion definition.

In this example any Parameter name that starts with "BLOCK\_TEST" will be ignored.

Note that in the second Exclusion definition the name has the '.' suffix character indicating that any Parameter name that starts with "UPDATE\_EVT." will be ignored. Use of the hierarchical delimiter char '.' helps to isolate Parameters that should be ignored. If a Parameter name "UPDATE\_EVTOP.OPSYNC" exists, then this Exclusion definition would not apply based on the position of the '.' delimiter character. However a Parameter name "UPDATE\_EVT.INSYNC" would be excluded.

<b>J</b>	BLOCK_TEST*	Block	DEVFB	
<b>J</b>	LIDDATE EVT *	Block	DEVER	



### Scope column



Specify the configuration scope where this Parameter exclusion applies.

You can choose to exclude a Parameter name that is found in a Function Block, Control Module, or both Function Block and Control Module.

Specifying Control Module as the scope option will only exclude a Parameter name if it is found in what is considered the Control Module definition block of a CM. This Control Module block contains the configuration data that is not specific to any Function Block that is embedded in the Control Module container. Examples of Control Module block type parameters would be SCANASSOCDSP (Associated Display), DESC (CM description), EUDESC (CM EU descriptor), etc.



### ClassName column

ClassName

Specify an optional ClassName to further identify that this Parameter is only to be excluded if found within a Block that has the specified Class Name type.

In the example here the ClassName value "DEVFB" refers to CEE FieldBus device type Function blocks. Other common ClassName values would include FLAGNIM, NUMERNIM for LCN type entity configuration.



Specify and optional BlockName to further identify that this Parameter is only be excluded if found within a Block that has this BlockName value.

In this example below, the BlockName value of "PID" indicates that any Parameter found in a Block whose BlockName value is "PID" will be excluded (ignored) during value comparison. The wild card character "\*" value for the ParamName indicates that all Parameters apply for this exclusion definition.

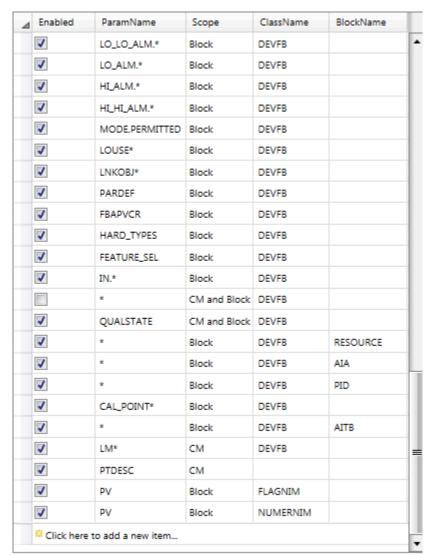


# Parameter Exclusion tab



Click this tab to show the Parameter Exclusion View.

## Parameter Exclusion list



The list of defined Parameter exclusions. Configuration data often contains Parameter values that are expected to change during runtime such as FLAG block PVFL status, field device runtime parameters, etc. Choosing to define these parameters as excluded will help to eliminate noise in the generated comparison report.

# Save button



Click this button to save changes made to the list of Parameter exclusions.