CygNet v9.4 Release Notes

Release Date: February 28, 2020

This document describes new features and changes to CygNet Software since the v9.3 release. For instructions on updating your host, refer to the *CygNet v9.4 Upgrade Procedure* document.

Contents

Product Lifecycle

For more information on the lifecycle of CygNet components, refer to the *CygNet Software Product Lifecycle Matrices* on the CygNet Support portal (login required).

CygNet Component Lifecycle Notice

The following products and components have reached lifecycle milestones with the release of CygNet v9.4:

- **CygNet v9.1** will enter Limited Support on July 20, 2020 and will reach its End of Life on July 20, 2021.
- **CygNet v9.0** is in Limited Support effective December 19, 2019 and will reach its End of Life on December 19, 2020.
- **CygNet v8.5.1** is in Limited Support effective June 16, 2019 and will reach its End of Life on June 16, 2020.
- **Windows 7** is no longer a supported operating system for CygNet v9.4.

  Windows Server 2008 R2 is no longer a supported operating system for CygNet v9.4.

Please contact your Account Manager or CygNet Sales (CygNetSales@weatherford.com) for more information.

Upgrade Assistance

Upgrade assistance is provided through prepaid professional service hours provided with your **Annual Maintenance Agreement** or through time-and-materials consulting services. If you need assistance in planning, upgrading, or deploying this release, please contact CygNet Support for more information about these options. CygNet Support can be reached at 1-866-4CYGNET (1-866-429-4638) or CygNetSupport@weatherford.com.

CygNet Documentation

The *CygNet Help* documentation for CygNet v9.4 is accessible from https://softwaredocs.weatherford.com/cygnet/94/.

Highlights in v9.4

This section highlights new features and enhancements in the v9.4 release. Please see Changes in v9.4 below for a detailed list of changes. Refer to the *CygNet Help* for user assistance on these enhancements, modifications, and other updates.

Canvas HMI Client

Many new and enhanced controls, configuration and scripting options, and improved usability features have been implemented in Canvas v9.4. See the full release notes section for more information about each major highlight listed below.

Performance is improved, and screen load time is significantly reduced in Canvas v9.4. Script assemblies are handled more efficiently resulting in faster load times by removing the need to recompile script every time a screen is loaded. Screen load time metrics are also provided. See the full note here.

A **Navigation by Facility** feature has been added to Canvas to allow you to navigate to a specific screen by selecting a CygNet facility. Several navigation modes are available depending on your desired usage. See the full note here.

The **Alarm Notifier** is an optional real-time interactive banner providing visual and audio notification of all active alarms that displays along the top of a screen viewed in Canvas View and Canvas View Lite. See the full note here.

Relative facility definitions and control configuration have been enhanced to support **ordinalized facilities**. See the full note here.

Canvas now supports custom **color palettes**, which can be used to override the default point state colors for any point type background and foreground color. Palettes are configured globally and applied to CygNet-aware controls. See the full note here.

A new read-only run-time companion application to Canvas is available, **Canvas View Lite**. Canvas View Lite utilizes a single main screen that drives all other screen navigation. It is always full screen and removes all user interaction that is not part of the screen’s HMI. Canvas View Lite is opened via command-line using the same options as Canvas View. See the full note here.

The **Button** is now CygNet-aware. Supported features include: CygNet point configuration; color configuration; blinking when the associated point is in alarm; configurable image size and resizing behavior; dynamic text using tokens; tooltips; context menu; font sizing, and a configurable mouse cursor. See the full note here.

The **Chart** now supports automatic live updates by requesting new data at a specified interval. Several new series types have been added, including a bar, line area (default), spline, spline, step, and step area. Charts now support the display of custom non-CygNet data via script. See the full note here.

Several enhancements have been made to the **Donut** to provide greater visibility into the alarm ranges for a point and to allow dynamic determination of the lower and upper value bounds. Alarm ranges are indicated on the inner ring and a configurable marker points to the current value. Supported features include: CygNet point configuration; color configuration; value bounds configuration; dynamic text using tokens; tooltips; context menu; and font sizing. See the full note here.

The **Alarm** and **CygNet Grids** screen performance has been improved to better handle loading, scrolling, sorting, and mouse actions. You can now export data from a grid column (except History) via script. See the full Alarm Grid note here and see the full CygNet Grid note here.

A new CygNet-aware **Horizontal Linear Gauge** has been added to Canvas. Like the vertical linear gauge this data visualization tool presents real-time point data in a bar between a lower and upper bound. The control supports both linear and logarithmic scales. Supported features include: CygNet point configuration; color configuration; value bounds configuration; dynamic text using tokens; tooltips; context menu; and font sizing. See the full note here.

Value bounds configuration has been added for the Horizontal and **Vertical Linear Gauge**. The source of the lower and upper value bound can be hard coded or dynamically determined for each facility. This configuration is also available in the Donut and Value Indicator controls. See the full note here.

The **Image** control is now CygNet-aware and supports dynamic image display based on the point state of the associated point. Supported features include: CygNet point configuration; color configuration; configurable image size and resizing behavior; dynamic text using tokens; tooltips; context menu; font sizing, and a configurable mouse cursor. See the full note here.

A new navigation container has been added, the **Navigator**, which can display any number of Canvas screens. The Navigator includes a hierarchical navigation menu, which can be configured to organize access to different screens. See the full note here.

A **Search Box** has been added to Canvas to search for facilities or other objects in your CygNet environment. You can search all facilities in a FAC service with three options available when a facility is selected from the search results: the search box acts as a facility sender; the search box navigates by facility and links to an associated screen; or another scripted action can be executed. You can also perform a custom scripted search against any object, for example, a SQL database, a .csv file, or a web site. See the full note here.

A new CygNet-aware control, the **Shape** control, is provided to draw geometric shapes or custom paths on your screens. Supported features include: CygNet point configuration; color configuration; dynamic text using tokens; tooltips; context menu; font sizing, and a configurable mouse cursor. See the full note here.

The **Sparkline** supports automatic live updates by requesting new data at a specified interval. Also supported are configurable x-axis and y-axis, and minimum and maximum y-axis values, allowing you to detect larger deviations from the norm. See the full note here.

A new navigation container has been added to Canvas, the **Tab** control, which incorporates a collection of tabs across the top of the control and a nested view linked to associated content. The Tab control is a lighter version of the new Navigator. See the full note here.

The **Text Tool** now handles point processing when displaying static text differently. A new SuppressDataRetrieval property has been added to prevent the control from retrieving CygNet data, even if the control is configured with valid point information. Use this if you want to store point information but don't want the control to automatically query services. This feature is also implemented in the Button, Image and Shape. See the full note here.

A new analog control has been added, the **Value Indicator**, which adds visual context to real-time point and alarm data. A data range is represented on a vertical bar that fills the control between an upper and lower bound. Alarm ranges are indicated on the bar and a configurable marker on the side points to the current value. Supported features include: CygNet point configuration; color configuration; value bounds configuration; dynamic text using tokens; tooltips; context menu; and font sizing. See the full note here.

All single-point CygNet-aware controls support configurable **tooltips**. Three modes are available: plain and/or tokenized text that resolves to point and facility attributes; an object file, which can receive its facility tag from the control showing the tooltip; or a script option that creates a custom tooltip. See the full note here.

Canvas now supports **Layers** to show and hide multiple controls on one layer without affecting controls on another layer. A new Layers pane is available where you can configure layers in design mode and at run time. Script methods to control layer visibility at run time are also available. See the full note here.

You can now **nudge** Canvas controls in design mode. Use the up, down, left, and right arrow keys to nudge a control by 1 pixel. Ctrl+Arrow moves a control by 10 pixels. Ctrl+Shift+Arrow moves a control by 100 pixels. See the full note here.

Several properties have been added to allow customization of the screen **layout grid** in design mode: Grid color, Grid height, Grid width, Show grid, Snap to grid, and Snap to items. See the full note here.

Enhanced Alarm Configuration (EAC)

CygNet now features an Enhanced Alarm Configuration, which provides the ability to either trigger an alarm (via configurable bit setting) or prevent an alarm from being triggered based on information from one or more points other than the original point being evaluated. The EAC configuration is available from the PNT Editor's Alarm Settings (Analog, Digital, Enumeration, String) pages. See the full note here.

CygNet Dispatch

CygNet Dispatch now offers a broader range of job scheduling options, including allowing you to include non-meter-specific jobs in schedules, display automatic calibration values in job reports, utilize separate post-calibration test standards if desired, and benefit from improved integration with CygNet Measurement and the FMS service. See the full list of CygNet Dispatch release notes here.

CygNet Measurement (FMS)

CygNet Measurement has added new capabilities throughout the product for editing and managing liquid data for Nodes supporting periodic device history data. This includes support for liquid periodic data balances in the Balance control, and support for gas and liquid periodic data in normalization views created to ensure uniform data handling and publishing.

The 9.4 version updates and renames both the Station report and Station Data CSV export to include the additional normalization view data for periodic devices; these expanded versions are now called the Normalization View report and the Normalization View Data CSV export, respectively, to better reflect the new functionality.

In FMS Explorer, processing improvements in the Dashboard control enhance performance and responsiveness in all Dashboard data views.

A number of other enhancements have been introduced to support FMS interoperation with CygNet Dispatch, when used to integrate field information results into your measurement data. These improvements include a new FMS “Accessory” Node type for auxiliary jobs, an added “Job” category of FMS validation rules, and extended functionality in the Job control, including reviewing calibration events, approving “verified” records without calibration changes, or deleting jobs as deemed necessary for administrative purposes.

CygNet Measurement includes many other improvements in this release. See the full list of CygNet Measurement release notes here.

CygNet EIEs

Remote devices have been enhanced to include the following new features:

- A new cvtF conversion method, **TimeString**, has been added to all EIEs to handle string timestamps. See the full note here.
- Changes have been made to the Configurable Data Group for several EIEs. See the full note here.
- Other enhancements, modifications, and fixes have been added to several remote device EIEs and device template files as described here.

Job Runner

Some options have been added on the **ScriptJob** level to allow you to select an output mode (synchronous or asynchronous) and whether to write output and errors to the log file. See the full note here.

Changes in v9.4

This section describes enhancements, modifications, and fixes to existing components in CygNet v9.4.

CygNet Clients

The following changes have been made to CygNet client applications in CygNet v9.4.

Canvas

Application-Wide Enhancements

- **Performance** is improved, and screen load time is significantly reduced in Canvas 9.4. When a screen is saved in Canvas, any script assemblies are compiled into a uniquely named .dll file and added to the .can file, so the script does not have to be recompiled every time the screen is run. When a screen is loaded in Canvas preview or Canvas View, the precompiled .dll is extracted to a temporary directory and the script assembly is loaded from disk. This results in faster load times by removing the need to recompile script. A new property **Unique name** (UniqueName) has been added to the screen, which is used when creating the assembly for the screen's script. Under most circumstances, this will not need to be changed and can be ignored.

  If you have an older screen with large amounts of script, open and save the file in Canvas v9.4, and the script assemblies will be compiled and added to the .can file. Next time you preview the file or open it in Canvas View, the screen should load much faster.

- Significant effort has been invested in the way Canvas and Canvas View render screens in run mode, resulting in greatly **improved screen performance** and **reduced load time**. The new run mode also allows you to dynamically add controls via script. This enhanced run mode is the default method when you hit F5.

  A new **Load time** popup is provided when previewing screens to give a more granular understanding of where the time is being spent as the screen loads. Load time is a summation of the time it took to process all the elements of the screen: initialize screen, parse xml, load controls, apply styles, run script, etc. The Load time popup is accessible from the Canvas Viewer toolbar when previewing a screen; it is not available in Canvas View. The load times are also available via the log file if Debug logging is enabled.

  One of the adverse side effects of the new run mode is that it does not support panning and zooming. Thus, we have retained the old method of running screens for backwards compatibility. If you need that feature on a specific screen (via the **Is pan/zoom enabled** (IsPanZoomEnabled) property), you will not be able to take advantage of the new features and performance improvements this new run mode supports. The legacy run mode is accessible from the **Run** icon on the Canvas toolbar.

- A **Navigation by Facility** feature has been added to Canvas to allow users to navigate to a specific screen by selecting a CygNet facility. You can also include optional facility record properties when the navigation is triggered. Several navigation modes are available depending on your desired usage:

  - hyperlink to a facility's associated screen in a new window
  - replace a current screen with a facility's associated screen (with or without a navigation toolbar)
  - notify any navigator controls on the current screen to replace the active screen with a facility's associated screen (with or without a navigation toolbar)
  - navigate by firing a screen script event

  Configuration for the **Navigation by Facility** feature is required in the Canvas global settings on the Backstage view, in the facility record in your CygNet system, and by selecting the desired facility navigation mode on each Canvas screen.

The navigation by facility feature is also available via script using the following APIs:

  - **FacilityNavigationMode** property
  - **NavigateByFacility** event
  - **OpenedViaHyperlink** event
  - **NavigateToScreenByFacility** method

Two other new features, the alarm notifier and the search box control, also use navigation by facility to link to a facility's associated screen using any of the navigation modes described above.

Refer to the **Scripting in Canvas** section in the *Canvas Help* for more information about **Navigation by Facility**, the **Alarm Notifier**, and the **Search Box** control.

- The **Alarm Notifier** is an optional real-time interactive banner that displays along the top of any Canvas screen or object viewed in Canvas View and Canvas View Lite. The notifier pulls its information from the Common Alarm Services (CAS) where centralized alarm processing for the current value services (CVSs) takes place. The alarm notifier is used to monitor and acknowledge alarms and view additional information about the facilities and points in alarm on related screens.

  The alarm notifier provides visual and audio notification of all active alarms and consists of two elements:

  - a box displaying the total number of unacknowledged alarms, which can be hyperlinked to a custom alarm screen

  - individual alarm boxes displaying each unacknowledged alarm and listing three configurable point or facility attributes and an ACK button for acknowledging the alarm. The top left attribute can be hyperlinked to a related screen for the associated facility.

  The navigation mode for both hyperlinks is determined by a property setting on the screen where the alarm notifier is displayed. See the Navigation by Facility note for more information about the navigation modes, configuration, and API that support this feature.

  The alarm notifier can be configured to play an alarm sound to signify unacknowledged alarms based on an alarm priority range, although alarm sounds can be disabled via a command-line parameter for the instance of Canvas View or Canvas View Lite. Alarms are displayed in priority order, can be filtered, and the background and text colors configured as desired.

- The following enhancements have been made to the Canvas **Relative Facilities** feature:

  - Relative facility definitions and relative links have been enhanced to support **ordinalized facilities**:

    - An **Order in Type** option is now available when configuring a definition within a definition set, to capture the facility attribute used to represent the order in type. This attribute can be different between definitions and is configurable at the individual definition level. A blank attribute value means that the property is not used.

    - For all Canvas controls that support relative facility resolution (including the screen/object), an **Order in type** property has been added to the Properties pane to specify the order in type/ordinal to use if the relative link is configured for one.

    - Relative facility definitions are created in the Canvas Backstage view and stored in the Canvas global settings file. Relative facility resolution (the desired link and ordinal) is configured for each screen, object or control configured with either a point or facility.

  - A **Relative Facility Tester** has been added to the **Relative Facility Definitions** interface in the Canvas Backstage view to test that a relative facility link resolves to the appropriate facility tags in the CygNet environment.

  - A **Relative Facility Definitions Diagram** tool has been added to the **Relative Facility Definitions** interface in the Canvas Backstage view to show the graphical relationship between the relative facility definitions within a definition set.

- Canvas now supports **Layers** to show and hide multiple controls on one layer without affecting controls on another layer. Each control can be assigned to a layer via the Layers property. A new Layers pane is available where you can show or hide layers on a screen in design mode and at run time. Layer visibility can also be controlled at run time via the **AddLayer**, **HideLayer**, and **ShowLayer** methods on the Screen object.

- The Canvas **plugin location** requirement has been relaxed in v9.4. Now a custom plugin assembly can be located in any Windows file-system folder accessible to Canvas. Previously all plugins had to be collocated with the Canvas executable in the **Bin** directory.

- The **Canvas v9.4 Online Help** is now available online at https://softwaredocs.weatherford.com/cygnet/94/ as part of the larger CygNet Online Help site. From within the Canvas application you can access the online help in the following ways: press the **F1** key, click the **Help** icon on the Canvas toolbar, or click the **Launch help** button on the Backstage view > Help. The CygNet v9.4 Online Help now includes the following user assistance resources: Canvas Online Help, CygNet Bridge API Online Help, and the CygNet Release Documents. The **Update Help** option has been removed from the Canvas Backstage View > Help page. Refer to the *CygNet v9.4 Upgrade Procedure* for instructions on removing the old Canvas Help files from your host and client computers.

Modification

- The Canvas **command-line interface** has been revised in v9.4 to improve usage and performance. Refer to the *Canvas Help* for details.

  - -Path parameter now works as expected
  - -Help parameter to provide user assistance has been added

Fixes

- Fixed an issue with Canvas that caused the **Toolbox** to appear empty and then the application would fail.

- Fixed an issue where invisible controls continued to take mouse input. Previously Canvas controls with the **Visible** (IsVisible) property set to False continued to take mouse input, so if they were stacked on top of other controls, the lower controls did not receive clicks or hover actions.

- Fixed an issue with **SharpVector** assemblies that was causing issues with web requests from Canvas.

- An issue adding new **plugins** to Canvas has been fixed. Previously selecting a plugin file from an **Open File** explorer dialog box caused an unknown error.

- Fixed an issue launching the **History > History Values** dialog box via a control's context menu in Canvas and Canvas View. Previously, the dialog box failed to display.

- Fixed an unexpected error encountered when attempting to create a **new style** from an existing style.

- Fixed an issue where the Type and Description column headers in the **Select Facility Tag** dialog box did not match the contents of the column. The headers have been swapped to match the column contents.

Canvas View

Modification

- The Canvas View **command-line interface** has been revised in v9.4 to improve usage and performance. Refer to the *Canvas Help* for details.
    - -FullScreen and -Path parameters now work as expected
    - -NoAudio parameter to suppress sound for the Alarm Notifier has been added
    - -Help parameter to provide user assistance has been added

Canvas View Lite

Enhancement

- A new read-only run-time companion application to Canvas is available, **Canvas View Lite**. Canvas View Lite utilizes a single main screen that drives all other screen navigation. It is always full screen and removes all user interaction that is not part of the screen’s HMI (no file menu, toolbar, backstage view, status bar, etc.). The main screen is opened via command-line using the same options as Canvas View. Both visualization applications are valuable for different HMI configurations. Because Canvas View Lite doesn’t use the docking framework that Canvas View does, it loads faster, at the cost of a far simpler UI and simpler layout options. Canvas View Lite can only open Canvas (.can) files, but it does support nested CygNet Studio (.csf) files.

Canvas Controls

Enhancements

The following changes apply to all or many Canvas controls.

- Many Canvas controls now support **custom color palettes**, which can be used to override the default point state colors for any point type for a point scheme for background color and foreground color in individual CygNet-aware controls. In general foreground color applies to the text color on a control; while the background color applies to all other elements where color is represented (e.g., background area, borders, lines, bars, rings, markers, and fills). A custom color palette is configured in the Backstage settings and then applied to any screen containing CygNet-aware controls (via the **Color palette** (ColorPalette) property). Color palette settings are saved as part of the global settings file (.gsf). In Canvas v9.4 custom color palettes are supported in the following CygNet-aware controls: alarm grid, button, CygNet grid, detail, donut, linear gauges, shape, text tool, tile view, and value indicator.

- All single-point controls now support configurable **tooltips**. There are three modes for the contents of the tooltip:

    1. plain and/or tokenized text that resolves to point and facility attributes

    2. an object file, which can receive its facility tag from the control showing the tooltip

    3. a script option that fires a **TooltipOpening** event right before the tooltip opens creating a custom tooltip, which can show any desired information, for example, a value from an affiliated point, an unrelated data point, or even a CygNet note

    The duration and display delay of the tooltip are configurable via script.

- All Canvas controls now support proportional resizing when resizing a screen horizontally or vertically. A **Proportional** option has been added to the **Horizontal resize mode** and **Vertical resize mode** properties.

- All Canvas controls now support **nudging** in design mode. Use the up, down, left, and right arrow keys to nudge a control by 1 pixel. Ctrl+Arrow moves a control by 10 pixels. Ctrl+Shift+Arrow moves a control by 100 pixels.

- The **Loaded** script event (fired when the control is rendered and ready for interaction) has been added to all Canvas controls.

- The **MouseEnter** and **MouseLeave** script events have been added to all Canvas controls.

  - MouseEnter event is called when the mouse pointer is moved over (enters) a control
  - MouseLeave event is called when the mouse pointer is moved out of (exits) a control

Alarm Grid

Enhancements

- Screen performance (loading, scrolling, sorting, mouse actions) has been improved for the alarm grid in v9.4.

- Canvas supports **color palettes**, which can be used to override the default point state colors for any point type for a point scheme for background color and foreground color in the alarm grid. See the full note here.

- Alarm grid columns can now be configured to sort numerically via the **Sort numerically** property. In run mode clicking a column header moves between the three states: sort ascending, sort descending, and unsorted. A small arrow indicates the sort order. Any column with mixed string and numeric data (like the Value column of the alarm grid) will be sorted as a string.

- The **Export** method has been added to export data from a grid column. Supported formats include CSV, HTML, and Text. Export from historical columns is not supported.

- The **Focus** method has been added to set the focus to the control to allow further operations.

- New APIs have been added to the alarm grid to improve multi-row selection via script.

  - **SelectAll** method — If the **SelectionMode** property is *Multiple* or *Extended*, **SelectAll** will select all rows in a grid and **SelectedRows** will return the matching DataRow objects. **SelectAll** will not obviously change the selection if **SelectionMode** is *Single*.

  - **SelectedRows** property (read-only) — **SelectedRows** will be restricted to a single row if **SelectionMode** is *Single*. Otherwise **SelectedRows** can contain multiple rows to be retrieved via script.

Modification

- Column configuration in the alarm grid has changed to make **default columns** optional. When the grid is first added to a screen the column list will be empty and you can add columns as desired. Or you can optionally use a set of 11 default columns. Previously the default columns were included automatically when the control was first added to a screen. The individual column properties have not changed.

Fixes

- The alarm grid can now send facility tag information to a receiving control. Previously when the alarm grid was configured as a facility sender (via the **Facility sender mode** property) the grid failed to send facility information to the configured control.

- **Acknowledging** an individual blinking alarm (via the context menu in run mode) now acknowledges the single alarm. Previously acknowledging an alarm caused the entire alarm grid to stop blinking until the grid was refreshed.

- **Force clearing an alarm** (via the context menu in run mode) now automatically updates the alarm grid to remove the cleared alarm from the grid. Previously the alarm was cleared but the grid continued to display the alarm until the grid was refreshed.

- Fixed an **alarm cache** issue that caused Canvas to crash when viewing an alarm grid screen with an incorrect or non-operational CAS.

- The **DoubleClickCell** and **SelectedCellChanged** events now fire as expected.

Button

Enhancements

- The **button** control is now CygNet-aware and shares most of the same properties as other single-point controls, such as the text tool. Supported features include: CygNet point configuration; color configuration for a button's background, text, and border; text and border blinking when the associated point is in alarm; hiding a button if the point is invalid; date, time, and value formatting for text displayed on a button; a context menu from where you can view current value, acknowledge alarms, access history and configuration information, and add values to a default chart; font sizing, text styling, and text weighting; tooltips (text, objects, or scripted); a static label or dynamic text using tokens; suppression of data retrieval, even if the control is configured with valid point information. The **Button text** (ButtonText) property has been replaced with a dynamic **Text** (Text) property. Older screens will copy the value of **Button text** over to the new property when a screen is loaded.

- Canvas supports **color palettes**, which can be used to override the default point state colors for any point type for a point scheme for background color and foreground color in the button. See the full note here.

- The size of the image on a button and its resizing behavior are now configurable. The following properties are now available:

  - **Image size** (ImageSize) indicates size of the image on the button.
  - **Image size mode** (ImageSizeMode) controls how the image behaves when the button is resized. The options are to automatically resize the image relative to the size of the control or to fix the size of the image using the Image size value. The default value is Auto.

- You can now configure the cursor that will display when the mouse hovers over the button using the **Mouse cursor** (MouseCursor) property. Options include Arrow, Hand, Wait, or Help. The default mouse cursor is the Arrow.

- The **Focus** method has been added to set the focus to the control to allow further operations.

Chart

Enhancements

- The chart control now supports automatic **live updates** by requesting new data at a specified interval. If live updating is disabled, the trend is populated one-time only with data from the VHS. Auto updating gives the appearance that the chart is trending data forward at a real-time pace. Live updates can be disabled/enabled for the current chart in run mode via a new icon on the toolbar. Also, an icon has been added to the toolbar to allow manipulation of the current chart's date and time range in run mode.

- Several **new series types** have been added to the chart, including a bar chart, line area chart, spline chart, spline area chart, step chart, and step area chart. The line chart is the default type when adding a new series. You can add multiple series of different types to the same chart.

- Charts now support the display of custom non-CygNet data via script. A **Data source** (DataSource) property has been added at the series level to indicate whether native CygNet or custom scripted data will be used.

Fix

- Fixed an issue where the chart would display history data for a different facility. Previously if a chart was displaying valid history data for a given facility and the chart facility context was changed to a new facility that did not contain points with the specified UDCs, then the chart would continue to display the history data for the previous facility. Now the chart will clear the legend and series if there are no matching points.

Combo Box

Enhancement

- The following properties have been added to allow color and text configuration for the combo box:
  - **Background color** (BackgroundColor)
  - **Text color** (TextColor)
  - **Text style** (TextStyle)
  - **Text weight** (TextWeight)
- The **Focus** method has been added to set the focus to the control to allow further operations.

CygNet Grid

Enhancements

- Screen performance (loading, scrolling, sorting, mouse actions) has been improved for the CygNet grid in v9.4.

- Canvas supports **color palettes**, which can be used to override the default point state colors for any point type for a point scheme for background color and foreground color in the CygNet grid. See the full note here.

- CygNet grid columns can now be configured to sort numerically via the **Sort numerically** property. In run mode clicking a column header moves between the three states: sort ascending, sort descending, and unsorted. A small arrow indicates the sort order. Any column with mixed string and numeric data will be sorted as a string.

- The **Export** method has been added to export data from a grid column. Supported formats include CSV, HTML, and Text. Export from historical columns is not supported.

- The **Focus** method has been added to set the focus to the control to allow further operations.

- New APIs have been added to the CygNet grid to improve multi-row selection via script.

  - **SelectAll** method — If the **SelectionMode** property is *Multiple* or *Extended*, **SelectAll** will select all rows in a grid and **SelectedRows** will return the matching DataRow objects. **SelectAll** will not obviously change the selection if **SelectionMode** is *Single*.

  - **SelectedRows** property (read-only) — **SelectedRows** will be restricted to a single row if **SelectionMode** is *Single*. Otherwise **SelectedRows** can contain multiple rows to be retrieved via script.

Detail Control

Enhancement

- Canvas supports **color palettes**, which can be used to override the default point state colors for any point type for a point scheme for background color and foreground color in the detail control. See the full note here.

- You can now apply additional emphasis to the text displayed on the detail control. The **Text style** (TextStyle) property supports normal, italic, and oblique text. The **Text weight** (TextWeight) property supports normal and bold text.

Donut

Enhancements

- Several enhancements have been made to the donut to provide greater visibility into the alarm ranges for a point and to allow dynamic determination of the lower and upper value bounds:

    - A configurable value marker is now displayed on the donut to indicate the point value.

    - A second inner ring is now supported to show alarm ranges that correspond to the alarm setpoints for the associated point.

    - The color of the donut's colored elements can be explicitly configured or sourced from the point state of the associated point.

    - The source of the lower value bound and upper value bound can be either hard-coded to an explicit value, or dynamically determined for each facility by sourcing the value from an application-specific attribute of the associated point (using an Indexed field or a General data field). A donut on a templated screen now supports different bounds for each source facility. By using a point attribute to store the bounds (e.g., Indexed 1 for the lower value bound and Indexed 2 for the upper value bound), you can have a templated screen without having to script the control. Three methods (SetLowerAndUpperValue, SetLowerBoundValue, SetUpperBoundValue) are available to adjust lower value bound and upper value bound via script.

- Canvas supports **color palettes**, which can be used to override the default point state colors for any point type for a point scheme for background color and foreground color in the donut. See the full note here.

- You can now apply additional emphasis to the text displayed on the donut. The **Text style** (TextStyle) property supports normal, italic, and oblique text. The **Text weight** (TextWeight) property supports normal and bold text.

Edit Box

Enhancement

- The **Focus** method has been added to set the focus to the control to allow further operations.

Image Control

Enhancements

- The **image** control is now CygNet-aware and supports dynamic image display based on the point state of the associated point. The control can be configured to display images that will change as the point state of the configured point changes. The control now shares many of the same properties as other single-point controls, such as the button or text tool. Other supported properties include: point configuration; hiding the control if the point is invalid; suppression of data retrieval, even if the control is configured with valid point information; a static label or dynamic text using tokens; date, time, and value formatting for text displayed on the image; a context menu from where you can view current value, acknowledge alarms, access history and configuration information, and add values to a default chart; font sizing, text styling, and text weighting; tooltips (text, objects, or scripted); and a configurable mouse cursor when the mouse hovers over the image.

- You can now configure the cursor that will display when the mouse hovers over the image control using the **Mouse cursor** (MouseCursor) property. Options include Arrow, Hand, Wait, or Help. The default mouse cursor is the Arrow.

Linear Gauge (Horizontal)

Enhancements

- A new CygNet-aware **horizontal linear gauge** control has been added to Canvas. Like the vertical linear gauge this data visualization tool presents real-time point data in a bar between a lower and upper bound. The horizontal linear gauge supports both linear and logarithmic scales. A primary scale is displayed along the bottom of the gauge and a secondary scale is displayed along the top side. An optional static or dynamic text label based on point or facility attributes is centered along the bar. The color of the horizontal linear gauge's background, text label, the bar representing the value, and the empty bar portion can be explicitly configured or sourced from the point state of the associated point. Tooltips and a context menu are also supported.

- The **value bounds configuration** has been added for the horizontal linear gauge in v9.4. The source of the lower value bound and upper value bound can be either hard-coded to an explicit value, or dynamically determined for each facility by sourcing the value from an application-specific attribute of the associated point (using an Indexed field or a General data field). A gauge on a templated screen now supports different bounds for each source facility. By using a point attribute to store the bounds (e.g., Indexed 1 for the lower value bound and Indexed 2 for the upper value bound), you can have a templated screen without having to script the control.

- Canvas supports **color palettes**, which can be used to override the default point state colors for any point type for a point scheme for background color and foreground color in the horizontal linear gauge. See the full note here.

- Added a **Click** event to the horizontal linear gauge, which will fire when a user clicks once on the control with the left mouse button.

Linear Gauge (Vertical)

Enhancements

- The **value bounds configuration** has been enhanced for the vertical linear gauge in v9.4. The source of the lower value bound and upper value bound can be either hard-coded to an explicit value, or dynamically determined for each facility by sourcing the value from an application-specific attribute of the associated point (using an Indexed field or a General data field). A gauge on a templated screen now supports different bounds for each source facility. By using a point attribute to store the bounds (e.g., Indexed 1 for the lower value bound and Indexed 2 for the upper value bound), you can have a templated screen without having to script the control.

- Canvas supports **color palettes**, which can be used to override the default point state colors for any point type for a point scheme for background color and foreground color in the vertical linear gauge. See the full note here.

- You can apply additional emphasis to the text displayed on the linear gauge controls. The **Text style** (TextStyle) property supports normal, italic, and oblique text. The **Text weight** (TextWeight) property supports normal and bold text.

- Added a **Click** event to the vertical linear gauge, which will fire when a user clicks once on the control with the left mouse button.

Modification

- To accommodate the new horizontal linear gauge, some changes were made to the vertical linear gauge:
  - The **Show right scale** property is now called the **Show primary scale** (ShowPrimaryScale) property
  - The **Show left scale** property is now called the **Show secondary scale** (ShowSecondaryScale) property
  - Text specified in the **Text** property is now displayed centered under the vertical bar

Navigator

Enhancement

- A new navigation control has been added to Canvas, the **navigator**, which is a container that can be populated to display any number of Canvas screens and objects. Intrinsic to the control is a hierarchical navigation menu, which can be configured to organize access to different categories of information. The navigation menu supports a single nested level of navigation and when a top-level or sub-menu item is clicked a related screen will be displayed in the body of the control. Each top-level menu item can display an icon to the left of the menu’s label. The navigation menu includes an adaptive display mode designed to preserve screen real estate: it can be configured to minimize all menu items, display only the top-level menu icon, or display the icon and the label. See the tab control for a similar new navigation control.

Nested View

Fixes

- Fixed an issue resolving to a relative facility for point configuration. Previously relative facility configuration caused an unknown error.

- Controls on a screen in a nested view will now resize as expected. Previously, when loading a screen in a nested view and the controls on the nested screen were set to any of the horizontal or vertical resize modes (shift, expand, or proportional) resizing did not occur.

- A display issue with all nested screens has been fixed. Previously when loading a nested screen for a second time, the screen would be offset six pixels down and right.

Object Container

Enhancement

- The **Focus** method has been added to set the focus to the control to allow further operations.

Screen and Object

Enhancement

- Several properties have been added to allow customization of the layout grid in design mode:
  - **Grid color** (GridColor)
  - **Grid height** (GridHeight)
  - **Grid width** (GridWidth)
  - **Show grid** (ShowGrid)
  - **Snap to grid** (SnapToGrid)
  - **Snap to items** (SnapToItems)

Search Box

Enhancement

- A new **search box** control has been added to Canvas to search for facilities or other objects in your CygNet environment. Two search modes are available:

  Facility Search

  This mode will search all facilities in the specified Facility service and return any matching facility names. Three options are supported when a facility is selected from the search result:

  1. **Facility Sender** — the search box acts as a facility sender and sends the selected facility to any controls on the screen configured to receive it.
  2. **Navigation** — the search box navigates by facility and links to the selected facility's associated screen. The navigation mode is determined by the Facility navigation mode setting on the screen displaying the search box.
  3. **Script** — another action can be scripted to execute when a facility is selected from the search results.

  Custom Search

  In this mode the search box can be scripted to perform a custom search against any object, for example, a SQL database, a .csv file, even a web site. Further custom actions can be executed when selecting an item in the search results.

- The **Focus** method has been added to set the focus to the control to allow further operations.

Shape Control

Enhancements

- A new CygNet-aware control, the **shape control**, is provided to draw geometric shapes, such as ellipses, triangles, rectangles, polygons, lines, lines with arrowheads, and other custom paths on your screens. You can use one of several predefined shapes or line types, or specify a geometric syntax to draw a custom path or line style. A shape's fill color, text color, and line color can be explicitly configured or sourced from the point state color defined for the associated point. A dynamic text label is included and can be built using tokens, which can represent real-time CVS point properties, point configuration properties, and facility properties. The shape can be set to suppress data retrieval, even if the control is configured with valid point information. A context menu is available in run mode with access to CygNet point and alarm data, including alarm acknowledgment, current and history values, point and facility configuration, and trending point data on a default chart.

- Canvas supports **color palettes**, which can be used to override the default point state colors for any point type for a point scheme for background color and foreground color in the shape control. See the full note here.

- You can apply additional emphasis to the text displayed on the shape control. The **Text style** (TextStyle) property supports normal, italic, and oblique text. The **Text weight** (TextWeight) property supports normal and bold text.

- You can configure the cursor that will display when the mouse hovers over the shape control using the **Mouse cursor** (MouseCursor) property. Options include Arrow, Hand, Wait, or Help. The default mouse cursor is the Arrow.

Sparkline

Enhancements

- The **sparkline** now supports a **configurable x-axis**, a **configurable y-axis**, and **minimum** and **maximum y-axis values**, allowing the user to detect larger deviations from the norm.

  - The x-axis can be constrained to the range of values for the associated point or fixed to the configured time range (default) via the **Auto range x-axis** (AutoRangeX) property

  - The y-axis can be constrained to the range of values for the associated point (default) or fixed to a configurable minimum and maximum value via the **Auto range y-axis** (AutoRangeY), **Maximum Y value** (MaximumYValue), and **Minimum Y value** (MaximumXValue) properties

  This enhancement also applies to the sparkline that appears on a detail control, and in the History column of the CygNet grid.

- The sparkline now supports automatic **live updates** by requesting new data at a specified interval. If live updating is disabled, the trend is populated one-time only with data from the VHS. Auto updating gives the appearance that the sparkline is trending data forward at a real-time pace.

- The **Background color** (BackgroundColor) of a sparkline is now configurable.

Fix

- Fixed an issue resolving to a relative facility for point configuration. Previously relative facility configuration caused an unknown error.

Tab Control

Enhancement

- A new navigation control has been added to Canvas, the **tab control**, which incorporates a collection of tabs across the top of the control and a nested view linked to associated content. The tabs allow you to easily switch between different Canvas screens and objects within a single container. The path to the linked screen can be a local or network Windows file-system folder or a CygNet APPS or BSS folder. The tab control is a lighter version of the navigator.

Text Tool

Enhancements

- Canvas supports **color palettes**, which can be used to override the default point state colors for any point type for a point scheme for background color and foreground color in the text tool. See the full note here.

- A **Font size mode** (FontSizeMode) property had been added to the text tool to control how the text behaves when the control is resized. The options are to automatically resize the text relative to the size of the control or to fix the size of the text using the **Font size** value.

- You can now configure **Text trimming** (TextTrimming) and **Text wrapping** (TextWrapping) properties for the text tool.

    - Text can be trimmed at a character or a word boundary and an ellipsis is added to indicate that the text has been trimmed. No trimming is also an option.
    - Text can be wrapped with very long words broken over multiple lines or truncated. No wrapping is also an option.

- You can now apply additional emphasis to the text displayed on the text tool. The **Text style** (TextStyle) property supports normal, italic, and oblique text. The **Text weight** (TextWeight) property supports normal and bold text.

- Added a **Click** event to the text tool, which will fire when a user clicks once on the control with the left mouse button.

Modification

- The text tool now handles point processing when displaying static text differently. A new property **Suppress data retrieval** (SuppressDataRetrieval) has been added to prevent the control from retrieving any CygNet data, even if the control is configured with valid point information. Use this if you want to store point information but don't want the control to automatically query services for data. Tokens will not be replaced and point state colors will be ignored. If a text tool has a point configured, and Suppress data retrieval is enabled, Canvas won't do much point processing. The one exception is for point validity. If the text tool is configured to **Hide invalid tag**, Canvas will check if the current tag is valid, even if **Suppress data retrieval** is enabled. This will allow you to configure some static text that will dynamically hide in the case of a templated screen for a facility that doesn’t have that particular point. The **Label** (IsLabel) property, which previously prevented point processing when the text tool was set to a static string, has been deprecated, but is still supported via backwards compatibility. The **Suppress data retrieval** property also applies to the button, image, and shape controls.

Fix

- A **style sheet** issue when attempting to apply a fixed dimension to both height and width when only one property was applied has been fixed. Previously, if you created a new text tool style from a default text tool and customized the height property, when that style was applied as the default to any new text tools, Canvas modified both the default height and width of the control, which was unexpected.

Tile View

Enhancement

- Canvas supports **color palettes**, which can be used to override the default point state colors for any point type for a point scheme for background color and foreground color in the tile view. See the full note here.

Value Indicator

Enhancements

- A new analog control has been added to Canvas, the **value indicator**, which adds visual context to real-time point and alarm data.
    - Any configured alarm limits for the associated point are indicated on the bar and an animated marker on the left side points to the current value.
    - The normal range for the point is depicted using a light blue color. The color of the control's background, text label, marker border, and marker interior can be explicitly configured or can be sourced from the point state of the associated point. The alarm limit range can optionally be configured to display point state color.
    - A configurable static or dynamic text label based on point or facility attributes is positioned under the control.
    - A data range is represented on a vertical bar that fills the control between an upper and lower bound. The source of the lower value bound and upper value bound can be either hard-coded to an explicit value, or dynamically determined for each facility by sourcing the value from an application-specific attribute of the associated point (using an Indexed field or a General data field). Three methods (SetLowerAndUpperValue, SetLowerBoundValue, SetUpperBoundValue) are available to adjust lower value bound and upper value bound via script.
- Canvas supports **color palettes**, which can be used to override the default point state colors for any point type for a point scheme for background color and foreground color in the value indicator. See the full note here.
- You can apply additional emphasis to the text displayed on the value indicator. The **Text style** (TextStyle) property supports normal, italic, and oblique text. The **Text weight** (TextWeight) property supports normal and bold text.

Other CygNet Clients

The following changes have been made to the following CygNet client applications in CygNet v9.4.

CygNet Explorer

Fixes

- Fixed an issue in the **DDS pane** of CygNet Explorer when File > New is selected to create a new device. Previously, when selecting File > New in the DDS, CygNet Explorer would crash.
- Fixed a caching issue in CygNet Explorer where the application would crash when attempting to scroll through a large number of point values in the **History Values** dialog box.

CygNet Host Manager

Fix

- Fixed an issue in CygNet Host Manager so that a crash does not occur on failover log rollover. Previously, a crash could occur shortly after a new log file was generated.

CygNet Studio

Enhancements

- The **CygNet Broadcast** feature has been enhanced to optionally report all messages to the Audit Service. A new keyword, REPORT_TO_AUDIT has been added to the Gns.cfg service configuration file (default value is FALSE). After updating to v9.4, run the **CygNet Config File Manager** to update your service configuration files.
- In CygNet Studio, added an **Export** button to the **Point List** dialog on The Frame / The View in Run mode.

Modification

- Interprocess communication between nested views and their parent screens now works in the following way:
  - When a parent screen sends a message and the optional Boolean parameter is set to false (to not to send the message to itself), the message will not be sent to any nested views contained within the sending parent screen.
  - When a parent screen sends a message and the Boolean parameter is set to true, the message will be sent to any nested views contained within the sending parent screen.
  - When a nested view sends a message, the message will always be sent to the parent screen holding the nested view, regardless of the Boolean setting (true, false, or not specified).

Fixes

- Fixed an issue in the CygNet Studio **Group Grid Tool** so that cells in the grid no longer flicker when values in the grid are updated. Previously, when a large number of columns and rows were present in the grid and an update was made, flickering could occur.
- Fixed an issue with the **Group Navigation Bar Tool** that caused CygNet Studio to fail. Previously the tool would get stuck when traversing a complex Group Service hierarchy containing recursive loops.

- Calling the **ShowModalView** method from the Microsoft TabStrip control now works as expected. Previously the modal dialog box launched by the tab strip blocked all user interaction.

- Fixed an issue for CygNet Studio screens containing a **UIS Command Button** tool so that an error is not returned when the screen is in Run mode. Previously, on screens containing a **UIS Command Button**, an error would be generated every 5 seconds when the screen was placed into Run mode.

- Fixed an issue displaying floating-point numbers in the **Watchlist**. Previously if you edited the current value of a point in the Watchlist to a float number, the tool appended a string of random digits to the point value.

- Fixed an issue in CygNet Studio so that when the command-line parameter "xmledit" is run, an xml file of the referenced Studio screen is created in the same folder as the Studio screen. Previously, when the "xmledit" command was run, the xml file was not generated.

CygNet Vision

Fixes

- Fixed an issue in the CygNet Studio **Group Grid Tool** so that cells in the grid no longer flicker when values in the grid are updated. Previously, when a large number of columns and rows were present in the grid and an update was made, flickering could occur.

- Calling the **ShowModalView** method from the Microsoft TabStrip control now works as expected. Previously the modal dialog box launched by the tab strip blocked all user interaction.

Modification

- **Interprocess communication** between nested views and their parent screens now work in the following way:

  - When a parent screen sends a message and the optional Boolean parameter is set to false (to not to send the message to itself), the message will not be sent to any nested views contained within the sending parent screen.

  - When a parent screen sends a message and the Boolean parameter is set to true, the message will be sent to any nested views contained within the sending parent screen.

  - When a nested view sends a message, the message will always be sent to the parent screen holding the nested view, regardless of the Boolean setting (true, false, or not specified).

CygNet Clients Online Help

Enhancement

- The CygNet v9.4 Online Help is now available online at https://softwaredocs.weatherford.com/cygnet/94/ and is accessible by pressing the **F1** key or clicking the **Help** icon from the following clients:

  - CygNet Explorer
  - CygNet Studio
  - CygNet Vision
  - Canvas
  - CygNet Console
  - FMS Explorer
  - CygNet Reports
  - CygNet Dynagraph Viewer
  - BSS Explorer utility
  - MSS Viewer utility

The CygNet v9.4 Online Help now includes the following user assistance resources: Canvas Online Help, CygNet Bridge API Online Help, and the CygNet Release Documents. The **CygNet.chm** file has been removed from the APPS and is no longer available for download via the CygNet Client Installer. Refer to the *CygNet v9.4 Upgrade Procedure* for instructions on removing the old CygNet.chm from your host and client computers.

CygNet Bridge

The following changes have been made to CygNet Bridge in CygNet v9.4.

Fix

- Fixed an issue with CygNet Bridge, so that performance is improved when transferring larger files (greater than 64K), thereby preventing a potential service crash.

CygNet Dispatch

The following changes have been made to CygNet Dispatch in CygNet v9.4.

Enhancements

- Added support for a new job type value, "Auto-calibration" (driver-assisted calibration) to distinguish from manual calibration jobs in the Jobs control and Job report templates.

- Added an optional job value enumeration to calibration Job report templates to support using a separate test standard set for post-calibration data values.

- Made a number of enhancements in CygNet Measurement to support interoperation with CygNet Dispatch. See related notes here.

Modifications

- Made a database modification in CygNet Measurement to support Job reports for CygNet Dispatch. See note here.

Fix

- Fixed an issue with scheduled jobs so that previously deleted scheduled jobs can be re-scheduled and synchronized with CygNet Measurement successfully.

- Made a number of fixes in CygNet Measurement to support interoperation with CygNet Dispatch. See related notes here.

CygNet Measurement (FMS)

The following changes have been made to CygNet Measurement in CygNet v9.4.

Enhancements

- In support of **liquid editing** functionality, **Normalization View** definitions have been expanded to include both gas and liquid device Nodes with periodic history support, and the configuration dialog in the Administrative Options has been enhanced to clearly assign and order available data items for a normalization view, including grouping data items by type to find them more easily.

- Enhanced and renamed the **Station report** to become the "Normalization View" report and provided a new sample template with the added ability to include normalization view data for gas or liquid device Nodes with support for periodic history data. Previously configured Station reports are still supported as configured, or can be updated using the new Normalization View report template if desired.

- Enhanced and renamed the **Station Data CSV export** to become the "Normalization View" Data CSV export with the added ability to include normalization view data for gas or liquid device Nodes with support for periodic history data. Previously scheduled Station Data CSV exports are still supported as configured.

- Improved processing in the **Dashboard control** to increase performance for all data views in the control.

- In the **Balance control**, added support for Liquid Device Nodes with support for periodic metering data.

- Modified **Station Group Node membership** requirements so that virtual stations can now contain Liquid Device Nodes and be included in balancing.

- Added **normalization** options to Liquid Device Node properties, for Nodes that support periodic metering data, to facilitate the publishing of SCADA data using normalization views and to provide information used for balancing purposes.

- Added support for new data item tokens to the **Export: Device Data CSV command**, so that gas analysis liquid content values can be included in the export definition file (.edf) as needed.

- Added **Mass tolerance** parameters to the **Changed Device Data report**, to apply to gas and liquid device Node data.

- In the Administrative Options, modified the Custom Device Data configuration page so that **mass-weighted average** is now available as an aggregation method option for custom liquid metering process variables.

- In FMS Explorer, added the ability to optionally specify an end date when **purging data** so that more recent good data can be retained if desired.

- Added support for Batch **Flow Time** data so that it can be viewed in the Raw Data control, published in Batch reports, and exported using the optional Export: Flow-Cal CFX Data command.

- Added columns to the **Command Logs** view in FMS Explorer so they are available to display Submitted Date/Time, Executed Date/Time, and Completed Date/Time.

- For users of CygNet Dispatch, added a new **Accessory Node** type to CygNet Measurement, so that jobs can now be scheduled and reported for non-meter-specific components and activities in a facility (e.g. compressors, battery banks, maintenance, etc.).

- For users of CygNet Dispatch, added the ability to the FMS Explorer Jobs control to review event-based calibration values using a new **Review calibration events** option available via the control's context menu.

- For users of CygNet Dispatch, added a Job status value of "Verified" to the **Jobs control** so that calibration or auto-calibration type jobs can be clearly identified as being within a known tolerance and available for review and approval, for cases where no calibration was performed therefore no history modifications are necessary.

- For users of CygNet Dispatch, added the ability to the **Jobs control** for administrators to multi-select jobs in the grid and "Approve" recalculations or "Change Effective Date" settings for multiple jobs in one action.

- For users of CygNet Dispatch, added the ability to delete jobs from the **Jobs control**, so that administrators can remove irrelevant jobs, such as bad calibrations, and their related artifacts from the system.

- For users of CygNet Dispatch, added a new "Job" category to FMS validation rules. New **FMS Job validation rules** with "Calibration Point" rule types are now available to validate that the number of calibration points present in the selected job report (for differential pressure, static pressure, or temperature calibration values) matches a specified number of points.

- For users of CygNet Dispatch, added new **FMS Job validation rules** with "Calibration Drift Threshold" and "Calibration Drift Range" rule types available to validate (for differential pressure, static pressure, or temperature calibration point values) that the drift threshold is within the specified high or low limit, and that the drift range is within the defined range, for the specified number of contiguous records.

- For users of CygNet Dispatch, added parameters to the **Job report** template to allow the use of automatic calibration values in the report if desired.

Modifications

- Modified the **Gas Device Raw QTR report** so that the [GasQualitySource] token displays a null value in the report when there is no live GQ source present.

- For users of CygNet Dispatch, modified the database to increase the maximum number of characters to 255 for text-based job report fields.

Fixes

- Fixed an issue with processing historical data, so that records are normalized as expected when retrieved data extends past the device timestamp.

- Fixed an issue with the **Import: Gas Quality CSV** command so that the import is successful when the default "Device" column header is replaced by a user-defined value.

- Fixed an issue with the **Export: Device Data CSV** command so the gas analysis HVV value reported for a periodic record correctly uses the value for the currently effective span.

- Fixed an issue with running the **Export: Device Data CSV** command for normalized data, so that live gas quality values are properly exported when available for device Nodes configured to support live gas quality history.

- Fixed an issue with the optional **Export: Flow-Cal Data CFX** command, so that exports including devices with large numbers of Liquid Product records are processed much more quickly.

- Fixed an issue with exporting liquid periodic data via the optional **Export: Flow-Cal Data CFX** command, so that only raw data is included in the export, and no user or system sourced data is sent.

- Fixed an issue with **FMS reports** so that uniqueness in enforced for report names, even when they originate from different report-building commands.

- Fixed an issue when saving **QTR reports** to PDF (Gas Device QTR, Gas Device Raw QTR, Liquid Device QTR, and Liquid Device Raw QTR), so that extra values are not inserted in the Pulse/DP column and subsequent records are not offset.

- Fixed an issue with the **Gas Device QTR report**, so that the gas analysis HVV value reported for periodic records correctly corresponds to the *_Begin or *_End token variant specified in the report template.

- Fixed an issue with the **Device Total QTR report** so that, when using token variants appended with "_End" for GQ data, the report is built without failing.

- Fixed an issue with the **Changed Device Data report** so that the volume tolerance parameters are observed correctly for liquid device Nodes.

- For users of **CygNet Dispatch**, fixed an issue with the calculation and application of "Effective date" values used when scheduling jobs, so that it occurs only for Calibration job types, and not for other types.

- For users of CygNet Dispatch, fixed an issue so that deleting a Job report template file from the Report Template folder occurs correctly, and a confirmation detail message appears as expected once Dispatch is synced.

- For users of CygNet Dispatch, fixed an issue to ensure that new jobs utilize the most recent report template when template modifications have been made.

- For users of CygNet Dispatch, fixed an issue with jobs in CygNet Dispatch and the FMS Explorer Jobs control, to ensure that cells are no longer editable once a job has been completed.

- For users of CygNet Dispatch, fixed an issue with the Jobs control so that copying jobs maintains the defined report command name.

- For users of CygNet Dispatch, fixed an issue with the Jobs control so that copying jobs converts time increments correctly in circumstances where the time zone and/or DST settings vary.

- For users of CygNet Dispatch, fixed the Sync results summary message to reflect the correct actions when jobs are synchronized with FMS.

- For users of CygNet Dispatch, fixed an issue with configuring Schedule Jobs command tasks in the MSS, so that the multiple character wildcard variable (*) correctly includes all defined jobs.

- For users of CygNet Dispatch, fixed a problem with scheduling jobs, so that the Schedule Jobs command is successful if synchronization with CygNet Dispatch yields already-completed jobs.

CygNet Services

The following changes have been made to CygNet services in CygNet v9.4.

Changes Affecting All CygNet Services

Fix

- Fixed an issue for all services so that a crash does not occur when LOGFILE_MODE is configured as EXTENDED. Previously, a crash could occur in this situation shortly after a new log file was generated.

Change Affecting Many CygNet Services

Fix

- Fixed a Facility and Point caching issue. Previously, under specific circumstances, some client-cached facility and point properties might not get updated, resulting in various applications, clients, and services displaying stale values. This condition could occur on systems with multiple CVSs where one CVS name is a substring of a second CVS and the second CVS includes a non-alphabetical character (for example, CYGNET.UIS and CYGNET.UIS1). Please note that a system with multiple CVSs with names such as CYGNET.UIS and CYGNET.UISA was not impacted by this caching issue. Point values and status bits (anything stored in the CVSs) would always be accurate for CVSs with these types of names.

Access Control Service (ACS)

Enhancements

- In support of the new Enhanced Alarm Configuration (EAC) feature, new security events have been introduced that control user access to the viewing and editing of EAC settings. Adding these new events is optional but, in their absence, user access to the EAC settings will mirror any existing configurable-bit permissions.

  The new "EAxxdtps" security event allows for point data type and point scheme qualification for a specific configurable bit. The new "EACALL" event has been added, which determines user access for all EAC settings regardless of the configurable-bit number, point data type or point scheme assigned. In addition, because the behavior of the new EA-based security events differs from that of the existing configurable-bit security permissions (e.g., CFGBIT03), a new , configurable-bit-specific security event (CBxxdtps) has been added for consistency. Refer to the **Enhanced Alarm Security** topic in the *CygNet Help* for more information.

Current Value Service (CVS)

Enhancement

- A new **Enhanced Alarm Configuration** (EAC) is now available in CygNet. The EAC allows a user to extend the conditions by which a configurable status bit is set (or prevented from being set) and any subsequent generated alarms, based on additional, logical expressions referencing real-time values of the same point or multiple other points. The EAC configuration is available from the PNT Editor's Alarm Settings (Analog, Digital, Enumeration, String) pages. New security events have been introduced that control user access to the viewing and editing of EAC settings. See the ACS for more information. All PNT databases must be re-indexed when upgrading to CygNet v9.4 to support the new EAC features. Refer to the *CygNet v9.4 Upgrade Procedure* for the steps to re-index the PNT service.

Device Definition Service (DDS)

Fixes

- Fixed an issue when copying devices in the DDS so that the **Device** button in the **Facilities Properties** dialog box is enabled. Previously, when copying a remote device and specifying a new facility, the **Device** button was not enabled after committing the change.

- Fixed an issue in the DDS so that when copying a remote device and editing the New facility ID field, a facility is created using either the Commit changes or Edit new device options. Previously, when choosing the Commit changes option, a new facility would not be created.

General Notification Service (GNS)

Enhancement

- The **CygNet Broadcast** feature has been enhanced to optionally report all messages to the Audit Service. A new keyword, REPORT_TO_AUDIT has been added to the Gns.cfg service configuration file (default value is FALSE). After updating to v9.4, run the Config File Manager to update your service configuration files.

Modification

- The keyword for the user password for the email server in the GNS service configuration file, EMAIL_PASSWORD, can no longer be decrypted using the Config File Manager. The encrypted keyword can be changed (but not decrypted) using the Config File Manager and the GNS Configuration Tester Utility.

HyperPoint Scripting Service (HSS)

Fixes

- Two methods have been added to the Job Runner to kill script processes:

  - KillAllScriptJobs()
  - KillScriptJob(int jobId)

  Previously the HSS was not able to restart successfully if a Job Runner script was still executing on the same port. To avoid this issue when the HSS shuts down, the scripter will need to make use of the **OnTerminateEx** event in conjunction with one of the new methods to either selectively terminate long-running scripts jobs or terminate all outstanding script jobs.

  Installation Instructions

  1. Make sure that Job Runner is not running in any process (for example, in CygNet Studio or the HSS; it will likely be one of these two processes but is not limited to these two).

     a. Close any open CygNet Studio/Vision clients.
     b. Shut down any HSS services hosting Job Runner scripts—be sure that all cscript.exe processes are stopped or manually terminate them.

  2. Version-manage CygNet Studio using CInstall.
  3. Restart any services stopped in step 1.

OPC Interface Service (OPCIS)

Fixes

- Fixed an issue for an OPCIS point where the alarm status bits would be cleared when setting the current value of the point to UNRELIABLE, such as when losing the connection to the OPC server. Subsequent retrieval from the OPC server generated additional alarms rather than keeping the initial alarm in place. Alarm status bits will now persist when setting the current value of an OPCIS point to UNRELIABLE.

- Fixed an issue in the OPCIS to handle a format string exception in the OPCIS. Previously, this exception was causing the reading of data to be delayed for the OPCIS.

Point Service (PNT)

Enhancements

- A new **Enhanced Alarm Configuration** (EAC) is now available in CygNet. The EAC allows a user to extend the conditions by which a configurable status bit is set (or prevented from being set) and any subsequent generated alarms, based on additional, logical expressions referencing real-time values of the same point or multiple other points. The EAC configuration is available from the PNT Editor's Alarm Settings (Analog, Digital, Enumeration, String) pages. New security events have been introduced that control user access to the viewing and editing of EAC settings. See the ACS for more information. All PNT databases must be re-indexed when upgrading to CygNet v9.4 to support the new EAC features. Refer to the *CygNet v9.4 Upgrade Procedure* for the steps to re-index the PNT service.

- A new engineering unit, "decatherms per hour" (DthH) has been added as an Energy Rate unit in the PNT Service.

Fix

- Fixed an issue on the **Analog** page of the PNT Editor so that when units are not indicated on the **General** page, the **Unit conversion** option is disabled. Previously, a user could select options in the **Unit conversion** menu when no units had been input on the **General** page, which caused CygNet Explorer and CygNet Studio to crash.

Remote Service Manager (RSM)

Fix

- Fixed an issue in the RSM so that a crash does not occur on failover log rollover. Previously, a crash could occur shortly after a new log file was generated.

Table Reference Service (TRS)

Enhancement

- The UDC description for the system UDC SYRSPEND (Pending Comm Queue Cnt per Remote Device), which is the number of messages the specified remote device has pending in its associated communication queue, has been added to the Table Reference Service (TRS) and to the *CygNet Help*. Perform the "Add New Table Entries to the TRS" step when upgrading to v9.4 in order to add this description. Refer to the *CygNet v9.4 Upgrade Procedure* for more information.

Universal Interface Service (UIS)

Fixes

- Fixed an issue in the UIS so that the UIS now logs a rejection message rather than crashing when a command request is received from a non-associated service.

- Fixed an issue for the UIS **View Pending Queue** dialog box so that **Device Name** and **Command Parameters** columns display correct data. Previously, there was a possibility of errant data displaying for these columns.

- Fixed an issue with VHS (Value History Service) logging behavior so that log results are now consistent with the values configured in the service configuration file (Vhs.cfg) for the Logging Info keywords LOGMASK_ELS and LOGMASK_FILE.

Value History Service (VHS)

Fix

- Fixed an issue in the UIS so that the UIS now logs a rejection message rather than crashing when a command request is received from a non-associated service.

EIEs – Communication Devices

The following changes have been made to CygNet communication devices in CygNet v9.4.

MQTT Comm EIE

Fix

- Fixed an issue with the MQTT Comm EIE so that redundancy failover attempts complete normally. Previously, redundancy failover attempts involving the MQTT Comm EIE would time out before completion.

EIEs – Device Template Files

The following change has been made to CygNet device template files in CygNet v9.4.

***Important:** If a device template file has been updated for this release, we strongly recommend that you obtain the applicable v9.4 sample device template file, edit it to retain customizations you added to your pre-v9.4 in-use template, and replace your pre-v9.4 in-use template with the version v9.4 sample template. Do not simply replace your pre-v9.4 in-use template with that provided on the source image because you will lose any template customizations that you previously made. Refer to the **Device Template Files** sections of the CygNet Help for detailed information about modifying templates.*

EIEs – Import/Export Devices

The following changes have been made to CygNet import/export devices in CygNet v9.4.

General Contract Monitor

Enhancement

- A new calculation for Accumulated Delta Percentage (AccumDeltaPercentage) has been added to the CygNet General Contract Monitor (GCM) calculation formulas. The result is a percentage of the daily accumulated value divided by the nomination value.

EIEs – Remote Devices

The following changes have been made to CygNet remote devices in CygNet v9.4.

Changes Affecting All EIE Remote Devices

Modification

- Added a new cvtF conversion method, **TimeString**, to handle string timestamps. The device template file includes a format specifier to define non-standard time formats, including ISO8601.

Fixes

- Fixed an issue so that a reference to a reference using the basic reference model is properly set for parent/children references on a data group element. Previously, a data group element using this type of reference model was not set correctly.

- Fixed an issue so that when writing reference data group elements using the advanced reference method, the correct order of operations is executed on both a Send and a Get. Previously, the order of operations would change for a Send.

Changes Affecting Some EIE Remote Devices

Modification

- For the "Configurable Data Group" (**CfgDg**) data group for the following EIEs, the template file attributes **canSend**, **uccSend**, **canRecv**, and **uccRecv** are used to determine whether or not each option is available on the **Configurable Data Group Definition** dialog box. For example, if canSend="false", the user will not have the option of setting it to true when configuring a data group instance.

    - AutoCom EIE
    - DNP3 EIE
    - DNP3 Emerson EIE
    - EProd EIE
    - IoT EIE
    - IoT Sparkplug EIE
    - OPC EIE
    - OPC Lufkin EIE
    - OPC Weatherford EIE

Allen Bradley CIP EIE

Fix

- Fixed an issue with the Allen Bradley CIP EIE so that a maximum string length of 470 is enforced by the EIE. Previously, requesting string tags where the string length was greater than 470 characters would result in an out-of-memory condition in the UIS.

Amocam 700 EIE

Modification

- The Amocam 700 EIE now allows record length overrides that are specified in the device template file using the "length" data group attribute. This override can be applied to these data groups:
    - "EGM Conversion Factors" (EGMConvF)
    - "EGM Factors" (EGMFacts)
    - "EGM Gas Composition" (EGMGas)
    - "EGM Parameters" (EGMParms)
    - "EGM Meter Run" (EGMMtrRun)
    - "Production History (26)" (ProdHist)
    - "Scan 21 - Analogs/Rates" (Scan21)
    - "Scan 22 - Analogs/Rates" (Scan22)
    - "Scan 2A - Rate/Vol/FlwCtrl" (Scan2A)
    - "Scan 2B - ATMOS Stn Scan list" (Scan2B)
    - "Tank Shutdown" (TankShut)
    - "Date, Time and Version" (TimeVer)

Benchmark EIE

Fix

- Fixed an issue for the Benchmark EIE so that data group element IDs (DEIDs) from reference items are included in "FMS Legacy History" (**GmrHist**) data group transactions. Previously, reference data group items could sometimes be omitted from "FMS Legacy History" data group transactions.

DNP3 EIE

Enhancement

- A new data group called "Configurable Data Group" (**CfgDg**) has been added. It provides a flexible way to create custom data groups on a per-device basis by using the remote device editor.

Fix

- For the DNP3 EIE, basic model method reference data group elements are now editable in the data group editor.

DNP3 Emerson EIE

Enhancements

- A new data group called "Configurable Data Group" (**CfgDg**) has been added. It provides a flexible way to create custom data groups on a per-device basis by using the remote device editor.
- For the DNP3 Emerson EIE, added the UIS command CLRHSTXA to clear all historical data group caches.
- For the DNP3 Emerson EIE, the deid "DateTime" is now used for **HistAlarm** and **HistEvent** transactions. Previously, "DateTimeE" was used, which caused the alarm/event timestamp to not display in the device editor.

Emerson ROC EIE

Enhancement

- For the Emerson ROC EIE, new FMS calibration Events have been added.

Fixes

- Fixed error handling during polling when a data group contains orphaned DEIDs that are mapped to points. Previously, this could result in a UIS crash.
- Fixed an issue when polling the Emerson ROC EIE History Point List (**HistPtCfg**) data group so that it is saved as a "Get Succeeded" or "Get Failed" DDS transaction after a poll. Previously, the transaction was saved as a "Save" rather than a "Get."
- Fixed an issue for the Configurable Opcode Settings data group of the Emerson ROC when the **Show numeric identifiers** checkbox is checked, the **TLP** and **Data type** columns display the numeric identifiers in parentheses. Previously, the numeric identifiers did not display when the checkbox was checked.
- Fixed an issue for the Emerson ROC EIE that caused the units in FMS data groups associated with liquid meters to occasionally be incorrect.
- Fixed an issue for the Emerson ROC EIE 107 device so that a **Send Gas Quality** command from FMS is successfully executed when the ordinal on the **FmsConfig** data group does not match the ordinal on the associated gas/liquid meter data group. Previously, if the **FmsConfig** ordinal referenced a different meter ordinal, a **Send Gas Quality** command would result in the data being sent to the wrong meter.
- Fixed an issue for the Emerson ROC EIE so that the DdsClient Method **GetDataGroupTxDataWithRefs** method now returns reference DEIDs from within the "Generic TLP Data" (**GenTLP**) data group. Also, the DEID names are also now returned from the GenTLP data group instance rather than from the device template.

Emerson ROCPlus EIE

Enhancement

- New FMS calibration Events have been added to the Emerson ROCPlus EIE.

Fixes

- Fixed an issue for the Configurable Opcode Settings data group of the Emerson ROCPlus EIE when the **Show numeric identifiers** checkbox is checked, the **TLP** and **Data type** columns display the numeric identifiers in parentheses. Previously, the numeric identifiers did not display when the checkbox was checked.

- Fixed an issue for the Emerson ROC Plus EIE so that user defined points are only assigned to User Programs that are installed on the device. Previously, the user defined points could be assigned to user programs that were not installed on the field device.

- Fixed an issue for the Emerson ROCPlus EIE so that a **Send Gas Quality** command from FMS is successfully executed when the ordinal on the **FmsConfig** data group does not match the ordinal on the associated gas/liquid meter data group. Previously, if the **FmsConfig** ordinal referenced a different meter ordinal, a **Send Gas Quality** command would result in the data being sent to the wrong meter.

- Fixed an issue for the Emerson ROCPlus EIE that caused the units in FMS data groups associated with liquid meters to occasionally be incorrect.

- Fixed an issue for the Emerson ROCPlus EIE so that when polling the "FMS Events" (**FmsEvent**) data group, the **Info** field of the polling result is populated correctly. Previously, some events incorrectly included "%s" instead of the correct description.

- Fixed an issue for the Emerson ROCPlus EIE so that the DdsClient Method **GetDataGroupTxDataWithRefs** method now returns reference DEIDs from within the "Generic TLP Data" (**GenTLP**) data group. Also, the DEID names are also now returned from the GenTLP data group instance rather than from the device template.

eProd EIE

Modification

- For the eProd EIE, added a new optional device-level attribute (alternateSOM) that directs the driver to look first for the start-of-message (SOM) character of 0x0A. If that fails, it then looks for the alternate SOM character defined in the template file.

Fixes

- Fixed an issue for the eProd EIE that reinstates the function that sets the DeviceClass = POC. Without this function, dynagraph cards could not be viewed. Previously, this override function had been removed from the device editor.

- Fixed an issue for the eProd EIE when retrieving alarm cards from field devices that require function code 16ext. Previously, alarm cards could not be retrieved from these devices.

- Fixed an issue with the eProd device driver when polling dynamometer cards from field devices with older firmware. Previously, ePIC controllers with firmware 1.10 were being polled using function code 16. Now function code 16EX is used for retrieving dynamometer cards for this firmware version.

- For the eProd EIE, added a new UIS Command component parameter ("InclPlain") for the "Dynagraph Card" (**DynaCard**) data group. When set to "true", a request for the most recent DynaCard alarm card returns the card associated with most recent Alarm or Plain event. When set to "false", it returns only the most recent Alarm event. Previously, a request returned only the most recent Alarm card. A user can now choose to see dynagraph cards associated with Plain events from the "Event Directory" (**EvtDir**) **Options** drop-down menu in the **View Data** dialog box.

IoT EIE

Enhancements

- For the IoT EIE, the **path** attribute in the **payloadDef** section of the device template file now allows the following substitution strings:
    - %deviceId%
    - %facilityId%
    - %ordinal%
- For the IoT EIE, the **itemId** attribute specified for a DGE now allows the following substitution strings:
    - %deviceId%
    - %facilityId%
    - %ordinal%
- The IoT EIE now allows users to write a payload containing a command, such as a request to republish data values. The "Command" (**Cmd**) data group can be issued manually through the data group editor or via a UIS command.

Modifications

- The IoT EIE now supports local time zones in payload files. In addition, time zones can now be configured on the remote device.
- The IoT EIE now supports multiple values for the same item identifier within the payload.
- The IoT EIE no longer has a requirement that **itemId** values be unique.

Fix

- The **itemId** attribute is now required in the device template file for the IoT EIE.

IoT Sparkplug EIE

Enhancements

- For the IoT Sparkplug EIE, the **itemId** attribute specified for a DGE now allows the following substitution strings:
    - %deviceId%
    - %facilityId%
    - %ordinal%
- The IoT Sparkplug EIE now allows users to request a rebirth manually through the "Command – Device" (**CmdDev**) or "Command – Edge Node" (**CmdNode**) data group using the data group editor or via a UIS command.

Modifications

- The IoT Sparkplug EIE now supports historical values for the same item identifier within the payload.

Fix

- The **itemId** attribute is now required in the device template file for the IoT Sparkplug EIE.

Lufkin SAM EIE

Modification

- Added alarm card retrieval and modified how failure cards are retrieved based on device firmware. Also added a new data group "Shutdown Log" (**SDLog**), which must be included in the device template file when retrieving the following card types:
    - **Alarm Card**: The device does not have a dedicated alarm card buffer. Instead, the Shutdown Log is queried. If the latest entry is for an alarm event, the alarm card is obtained from the shutdown card buffer.
    - **Failure Card**: The "Controller Version Information" (**CtrlInfo**) data group must be retrieved before requesting a failure card. The firmware version is used to determine which of two techniques to use to read failure cards.
        - In firmware version 5.28, Lufkin created a dedicated failure card buffer. If the device is using this firmware version or later, failure cards are obtained from the failure card buffer.
        - If the device is using an earlier version, the Shutdown Log is queried. If the latest entry is for a failure event, the failure card is obtained from the shutdown card buffer.
        - The version at which to use the dedicated failure buffer can be overridden in the template file by setting the **failureCardVersion** attribute in the" Dynagraph Card" data group.

  Refer to the Lufkin SAM EIE **Data Groups** topic in the *CygNet Help* for more information.

Totalflow EIE

Fix

- Fixed an issue for the Totalflow EIE so that a LOG_WARNING will not be recorded when a 6-byte value is returned instead of a 4-byte floating point value. In the past, the recommendation had been to reference the 4-byte register in the device template file instead of the 6-byte register. Some more recent Totalflow applications do not have a 4-byte floating point equivalent, so the warning has been removed.

Link Service

The following changes have been made to the Link Service in CygNet v9.4.

Enhancement

- Link now supports "Node Control/Rebirth" and "Device Control/Rebirth" commands for both Sparkplug and AWS configurations. Link will now issue complete birth payloads each time the service is started.

Fixes

- Fixed an issue where the Link message queue may not always drain to zero in very high latency environments when the quality of service is set to "Low" (MQTT QoS zero).

- Fixed an issue so that Link now publishes the first deid from DynaCard transactions for structured format messages. Previously, the first deid could be missing from the published DynaCard.

- Fixed an issue so that Link now includes **STime**, **STimeD**, **DTime**, and **DTimeD** in published Dynacards for structured format messages.

- Changed the trustedCertificatePath option in the Link security settings to "optional" so that users can use certificates installed to the Windows certificate store instead of a file path.

- Added an option (ignoreCertificateRevocationErrors) in the Link security settings to allow users to ignore errors from the certification revocation server.

Logging

The following fixes have been made to CygNet Logging in CygNet v9.4.

Fixes

- Fixed an issue for all services so that a crash does not occur when LOGFILE_MODE is configured as EXTENDED. Previously, a crash could occur in this situation shortly after a new log file was generated.

- Fixed an issue in the RSM so that a crash does not occur on failover log rollover. Previously, a crash could occur shortly after a new log file was generated.

- Fixed an issue in CygNet Host Manager so that a crash does not occur on failover log rollover. Previously, a crash could occur shortly after a new log file was generated.

Scripting

The following changes have been made to CygNet script libraries in CygNet v9.4.

CxPnt.PntClient

Fix

- The CxPnt.PntClient methods **AddPointRecord** and **UpdatePointRecord** now work as expected. Previously these methods would fail to add or edit a point record if the PointXML contained the "system_desc" point attribute and was set to anything other than "N/A".

CxScript.Points

Modification

- The **Points.AddPoint** method, which adds a point to the cache, has been modified to include two new optional tag parameters (TIMEONLY and ACKONLY). The syntax is as follows:

  ```
  Points.AddPoint "CYGDEMO.UIS:TANK_LEVEL;TIMEONLY", TANK_LEVEL_CMD
  ```

  The Tag string can include any of the following optional parameters to indicate that the Hyperpoint’s OnPointChange event will run only when the specific property changes, rather than running when any one of them changes:

  - ;VALUE – includes the base status, extended status, value, or alternate value. (Value is always included by default and matches currently released behavior)

  - ;TIMESTAMP – includes the value and the timestamp (This matches currently released behavior)

  - ;ACK – includes the value and the alarm record change (This matches currently released behavior)

    - ;TIMEONLY – includes only the timestamp (New)
    - ;TIMESTAMP – includes the value and the timestamp (This matches currently released behavior)

  The *CygNet Help* has been updated to clarify usage for this method.

Job Runner

The following changes have been made to the Job Runner in CygNet v9.4

Enhancements

- The following options have been added on the **ScriptJob** level to allow you to select an output mode (synchronous or asynchronous) and whether to write output and errors to the log file:
    - UseSynchronousOutput — This parameter controls when the output of a scripted job will be available for retrieval via the **GetJobOutput** method.
        - When set to true, the output will be available when the script job is complete, and there is no limit on the size of the output. The default value is true.
        - When set to false, the output is available while the job is running and upon job completion. Note that there is a limit on the size of output the scripted job can handle (between 4000 and 5000 characters).
    - WriteOutputToLogFile — This parameter controls whether the output is written to the log file. The default is false.
    - WriteErrorToLogFile — This parameter controls whether script errors are written to the log file. The default is true.

Fixes

- Fixed an issue with the Job Runner where script threads were blocking each other from starting. Previously when you started more jobs than the Job Runner was configured to support simultaneously, the remaining jobs were queued. All of the started jobs needed to finish executing before any of queued ones could start.
- Two methods have been added to the Job Runner to kill script processes:
    - KillAllScriptJobs()
    - KillScriptJob(int jobId)

  Previously the HSS was not able to restart successfully if a Job Runner script was still executing on the same port. To avoid this issue when the HSS shuts down, the scripter will need to make use of the **OnTerminateEx** event in conjunction with one of the new methods to either selectively terminate long-running scripts jobs or terminate all outstanding script jobs.

  Installation Instructions

  1. Make sure that Job Runner is not running in any process (for example, in CygNet Studio or the HSS; it will likely be one of these two processes but is not limited to these two).
      a. Close any open CygNet Studio/Vision clients.
      b. Shut down any HSS services hosting Job Runner scripts—be sure that all cscript.exe processes are stopped or manually terminate them.
  2. Version-manage CygNet Studio using CInstall.
  3. Restart any services stopped in step 1.

Utilities

The following changes have been made to CygNet utilities in CygNet v9.4.

ARS Diagnostic Tool

Enhancement

- The ARS Diagnostic Tool utility (ArsDiags.exe) has been enhanced to allow the addition of other domains so that you can monitor time synchronization across multiple domains in a redundant environment. A new **Add/Remove Domains** button has been added to the Time Test page to add and remove domains in an ad-hoc manner. Each domain shows its ARS information and related Cycle Time. A Summary section has been added to show Cycle Time, Earliest, Latest, and Delta values across all domains.

CygNet Config File Manager

Modification

- The keyword for the user password for the email server in the GNS service configuration file, EMAIL_PASSWORD, can no longer be decrypted using the Config File Manager. The encrypted keyword can be changed (but not decrypted) using the Config File Manager and the GNS Configuration Tester Utility.

Fix

- Fixed an issue in the CygNet Config File Manager (ConfigFileMgr.exe) so that the keyword values CMD_PRE_AUDIT_LIST and CMD_PRE_AUDIT_EXCLUDE_LIST are no longer saved with quotes. Previously, these keyword values would be saved with quotes, which could break the functionality.

CygNet DDS Command Copy

Fix

- Fixed an issue for the CygNet DDS Command Copy utility so that when a device is selected in the **Device ID to Copy** field, the selected device is correctly displayed. Previously, when the **Device ID to Copy** field was selected, the utility could crash or display an error message.

CygNet Service Migration

Enhancement

- Added a new **Update Option** to the CygNet Service Migration utility to include changes to template files for the OPC EIE, eProd EIE, and AutoCom EIE to specify that **canSend**/**canRecv**/**uccSend**/**uccRecv**=true for the "Configurable Data Group" (**CfgDg**).

DDS Check

Enhancements

- The DDS Check utility (DdsCheck.exe) now detects data groups that are no longer included in the device template file along with their associated UIS commands.

Point Configuration Manager

Enhancement

- New command-line parameters for edit (**/update**) and delete (**/delete**) functions have been added to the Point Configuration Manager utility (PointCfgMgr.exe).