Welcome to DeltaV™ Live

Table of Contents

Welcome to DeltaV Live

Table of Figures

Welcome to DeltaV Live

Introduction

DeltaV™ Live is Emerson's new, built-for-purpose operations experience, designed for today's high-performance operator requirements. It is a platform that is comprised of a suite of applications including a highly-customizable operator interface that will help you achieve operational excellence and an intuitive, state-of-the-art graphics development application called Graphics Studio. This document will discuss the many features of DeltaV Live and guide you in setting up your project library and associated displays.

DeltaV Live is a Thick Client Application for critical Operator consoles. The underlying technology of HTML5 Displays will soon be available on web browsers like Google Chrome or Microsoft Edge. Development is underway in DeltaV Mobile to support the DeltaV Live HTML5 displays

DeltaV Live includes numerous out-of-the-box features that follow industry standards like ISA 101.01 and industrial best practices such as those defined by the Center for Operator Performance (COP). These features allow users to design high performance graphics within a display hierarchy that adheres to these standards and avoids the need for custom scripting to achieve the desired results. Coordinated display navigation ("yoking") is supported as a standard feature, along with flexibility for Operators to create custom Watch Areas and integrate web-based external applications, creating an enhanced operations experience. For advanced applications, DeltaV Live offers the power of Type Script for configuring highly complex scenarios.
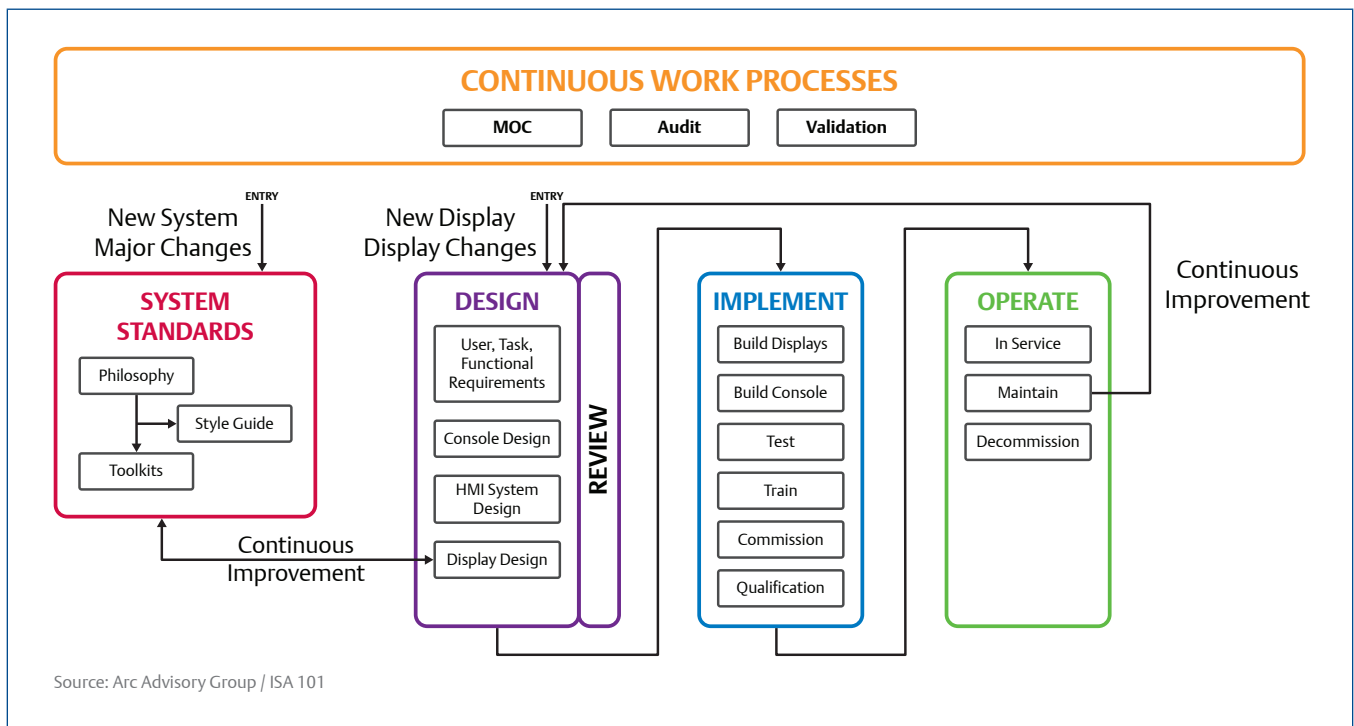

Figure 1. HMI Lifecycle Process

Welcome to DeltaV Live

To help you get started, this document focuses more on a discussion of the features of DeltaV Live rather than how to implement them. DeltaV Books Online is an important resource for your journey that provides additional details on how to create and work with the Graphics Studio application.

DeltaV Live Software

DeltaV Live offers a collection of applications to create, manage and visualize your process information. Each tool has been designed with Human Centered Design principles from the ground up to eliminate complexities and simplify workflows in order to enhance the overall user experience. The major tools of DeltaV Live can be summarized as shown below.
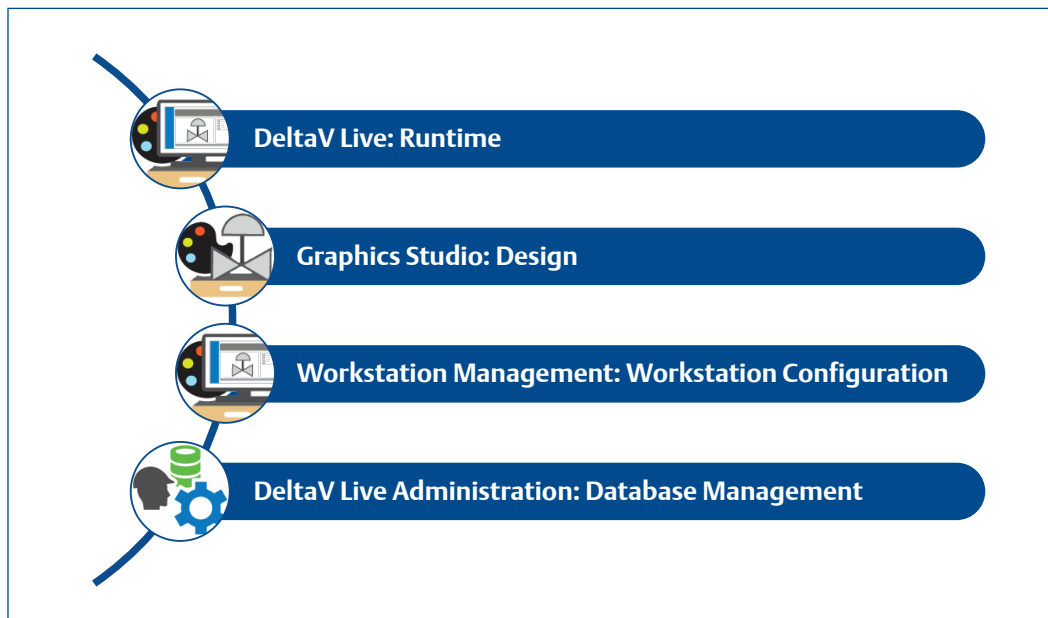
Figure 2. DeltaV Live Tools

DeltaV Live is the runtime (or online) environment for operators to monitor and control the process. Displays are published as HTML5 graphics and viewed in the purpose-built DeltaV Live environment, integrating the DeltaV alarm system, historical data, and real-time data into an operator environment that you design with Graphics Studio.

Graphics Studio is the configuration application where designing, building and previewing all aspects of graphics projects takes place. Graphics Studio includes libraries with ready-to-use graphical elements that speed up the process of engineering displays.

Workstation Management brings together the physical hardware components of each specific workstation with the functionality defined in Graphics Studio. In this application, users can assign layouts, display sets, and themes to specific workstations. Layouts help define how each screen will be used, and =display sets define display hierarchies for navigation. These are combined for each workstation running DeltaV Live.

DeltaV Live is based on a central configuration database. DeltaV Live Administration includes tools that allow the user to view DeltaV Live diagnostic information and manage graphics databases with actions that include create, delete, copy, export, import, or rename.

DeltaV Live Architecture

DeltaV Live display development is built on a relational database and takes advantage of the most sophisticated graphical technologies that are currently available. This is a significant departure from DeltaV Operate, which was a file-based system. It is important to understand some of these architectural differences and what they mean to your graphics project.

There are two main philosophies used by control system HMI architectures: server-and client-based architectures:

1. Server based architectures use a centralized server (typically redundant) to host displays and collect/serve runtime data to/from client stations.

2. Client based architectures use independent operator stations that communicate directly with the controllers, such that the operator stations are not dependent on a single server or intermediate node in order to perform their functions.

Both DeltaV Operate and DeltaV Live are designed around the latter, with each operator station being fully autonomous in performing its primary function. Centralized HMI servers are not used in DeltaV.

The similarities between DeltaV Operate and DeltaV Live basically end there. Although both can be configured to have a similar appearance at runtime, the underlying technologies are completely different. There are two main parts of the DeltaV Live architecture: the graphics design environment and the runtime environment. DeltaV Live displays are developed and stored in a central SQL database using Graphics Studio. This database contains Library components that constitute the basic building blocks for display creation based on design specifications defined by the end user. The runtime environment provides features consistent with the best practices defined in the ISA 101.01 standard and aligns with guidelines from the Center for Operator Performance.

Displays developed in Graphic Studio are published to a runtime database on each workstation, from which they are viewed as HTML5 displays without the need for web browsers. The use of HTML5 technologies enables DeltaV Live displays to be distributed outside of the operator station and serve DeltaV Mobile or Desktop clients that wish to view DeltaV Operator screens (note that this will be available in Release 4 of DeltaV Mobile).
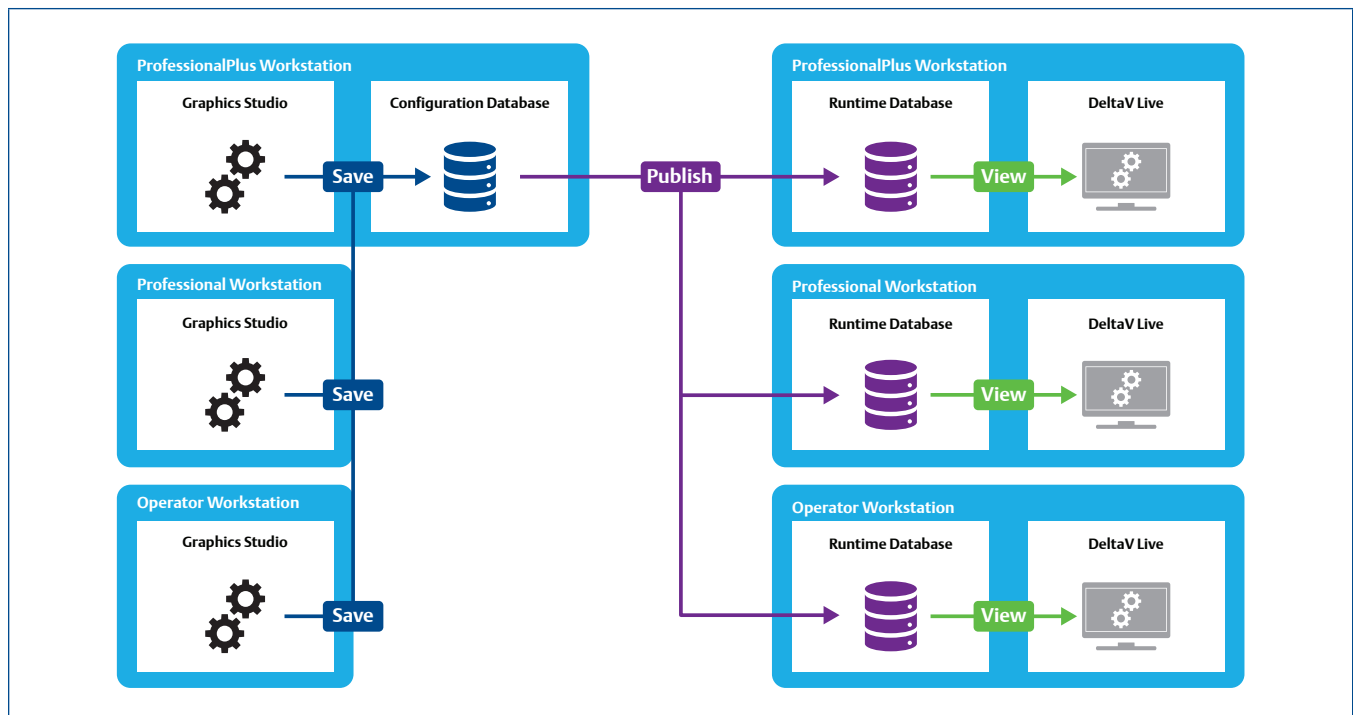

Figure 3. Configuration and Runtime databases

Welcome to DeltaV Live

Basically, DeltaV Live delivers the same autonomous workstation operation as DeltaV Operate but with enhanced engineering tools and management of change control. The Workstation Management application and distribution of displays via the new publish mechanism provides tighter control of when and where displays are distributed, eliminating the dependency on Yellow Pages.

Graphics Studio Interface

As we have learned, Graphics Studio is the application used to build graphics. It has been designed to facilitate the execution of a graphics project in adherence to defined display specifications. The ISA 101.01 HMI standard serves as the foundation for the architecture within the Graphics Studio. To that end, Graphics Studio features include the definition of a hierarchical navigation scheme aligned with multi-level displays. In addition, many Human Centered Design (HCD) features are provided out of the box. One of the features that stands apart from other HMI solutions is the ability to align the database Library with the best practices defined in your site's HMI standards.

Reusable components that can be used across the graphical project are organized in the Library Explorer:
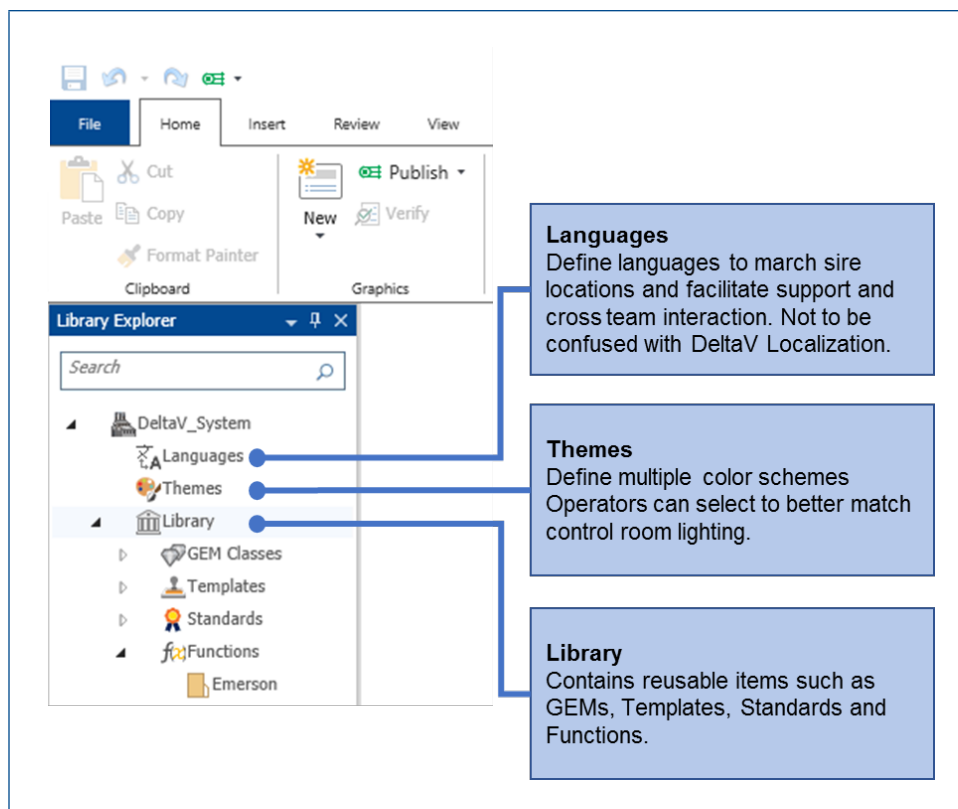
Figure 4. Graphics Studio Interface – Library Explorer

- **Languages:** DeltaV Live supports the use of multiple languages (up to 12) for all configurable string content that appears in DeltaV Live display configuration and online displays. This language-specific content allows different operators to see the same DeltaV Live display, in the same DeltaV system, with their preferred language content.

- **Themes:** Themes allow the same graphics configuration to appear differently on different workstations or at different times on the same workstation. For example, you can create a theme that minimizes eye fatigue in low-lighting control room environments, a theme that provides high contrast in brightly lit control room environments; or, an RGS (Reds and Greens Shifted) theme that aids operators who have difficulty distinguishing between certain colors.

- **Libraries:** Each graphical project can have multiple libraries; each library is independent from each other and contains the same groups of categories (See sub bullets). Emerson provides one library that contains many objects used for speeding up the design process. Users are encouraged to create their own libraries where they can modify and add items to implement a library that is compliant with their HMI specifications. Item names are unique within a library but can have the same name in different libraries.

  - **GEM Classes:** GEM classes are complex graphical elements used to represent process control equipment or other objects on a display. A GEM class provides a means to define a common appearance and behavior for all GEM instances that are linked to the class. GEM classes are managed as library items in configuration libraries. They are available in the Palette pane for you to add to displays or other GEM classes. A GEM class is added to a display or other GEM class as either a linked GEM or an unlinked GEM. Linked GEMs remain linked to their GEM class; therefore, when a GEM class is modified and saved, the changes propagate to all instances linked to that GEM class.

  - **Templates:** Graphics Studio allows creating templates for displays, contextual displays and layouts. These provide greater flexibility to the graphical project and speed up the design process.

  - **Standards:** Standards provide a means to ensure a consistent graphics appearance within a DeltaV system by providing the standard values to be used system-wide. They are created and managed in configuration libraries and can be referenced by properties in displays, layouts, or GEM classes. Graphics Studio provides a multitude of out-of-the-box standards including colors, fonts, Booleans, images, strings, measurements, and numbers. You can also create your own standards and assign a unique name within its configuration library.

  - **Functions:** Functions provide a means to create logic that converts values of one type into values of a different type. They are created and managed in configuration libraries and can be referenced by properties in displays, layouts, or GEM classes. You can create functions for float, string, and Boolean values to return values of various types, including colors, fonts, Booleans, images, strings, measurements, and numbers. The name of a function must be unique within its contained library.

Graphics Explorer groups the major components unique to the project being worked on, including:
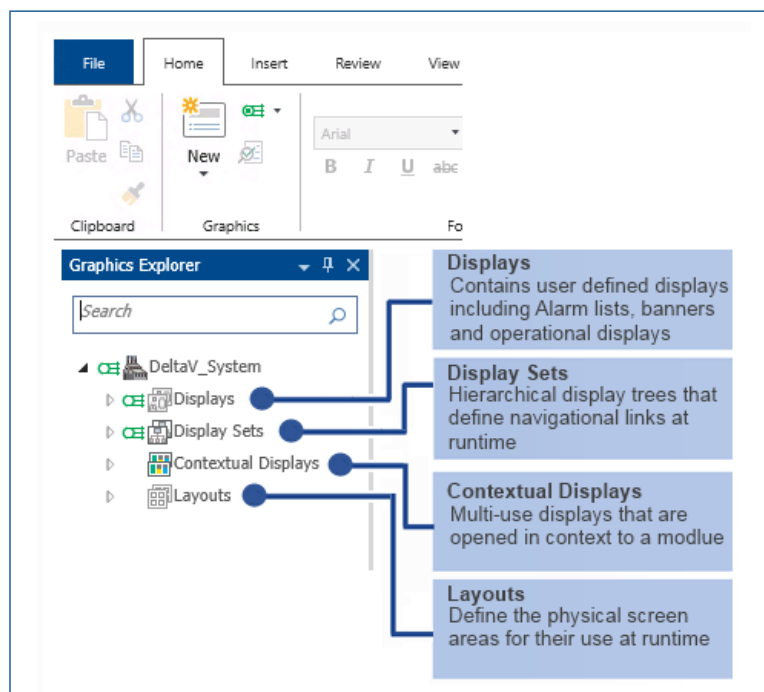

Figure 5. Graphics Studio Interface – Graphics Explorer

Welcome to DeltaV Live

- **Displays:** As the name indicates, these are the actual operator displays. They are all located under this section of the Graphics Explorer and can be organized in folders and subfolders to facilitate the overall project organization of displays.

- **Display Sets:** These define a collection of displays that match the responsibilities of the operators assigned to the workstation on which that display set will be available. This is designed to allow building display hierarchies consistent with industry standards like ISA101.01 by simply dragging and dropping displays into a hierarchical structure that automatically builds the navigation strategy for the user.

- **Contextual Displays:** A contextual display is a special type of display whose content varies depending on the context in which it is opened. Contextual displays provide the means to create and maintain a single graphics configuration that can be shared by multiple DeltaV objects.

- **Layouts:** Layouts arrange and purpose regions of an operator's available screen real-estate and define which displays should appear in each region. The display surface of a layout can be either a single screen (monitor) or an entire video desktop that includes multiple screens. Layouts also contain global variables, timers, and events.

Setting up your Library Explorer

Now that we've reviewed some of the highlights of DeltaV Live, the question likely to come up is where to begin. The first thing to understand is that the HMI development is somewhat iterative. You could start by creating a display, but you will need basic objects to add to the display. Will you be using single, dual, quad monitor stations? Or large 4K monitors, or some other number of monitors. Will you follow ISA 101.01 or implement your own set of standards and navigation scheme?

If your organization has a well-defined specification document for graphical projects, the first place to start is with the Library. Although DeltaV Graphics Studio delivers a functional Library Out-of-the-Box, it may not comply 100% with your organization's HMI Specification. It is recommended that folders are used within the Library to manage unique groups of items that are needed to fully develop your HMI specification.

> **Tip:** Folders help manage items in the Explorer tree, but references are only by Library name and Item name. You can move items between folders within a library without affecting their links to instances.

GEM Classes

The default Emerson Library created during installation is prepopulated with a set of GEM classes, templates, standards and functions. DeltaV Books Online provides more detail on the structure of the database, including themes and languages. This document will focus on recommendations and best practices and will refrain from duplicating the information from Books Online, except where it is expeditious to do so.

The out-of-the-box GEM classes are contained in a folder called Emerson, which contains seventeen sub-folders.

Figure 6. Out-of-the-box GEM Class folder structure

GEM classes in folders called "Components" are building blocks intended to be used to build more functional GEM classes that will be used in display building, such as the "High_Performance_Gems".

For ease of use, it is recommended to leave the out-of-the-box GEM classes as found and consider creating a new folder for adding project specific classes. You can also choose to move GEM classes from the default folders to your customized folder area as you use them in your project. In this way, you can have an optimized list of Classes to manage.

With this arrangement, unused or unmodified GEM classes are left in their original location. The new folder contains GEM classes that will be used in the project and these may be unmodified, modified, or new classes.

By grouping the used classes in a project folder, you can easily export these items in a single action.

You can add additional folders as needed, up to 4 levels deep. You can change the folder structure at any time as long as GEM classes remain in the same library.
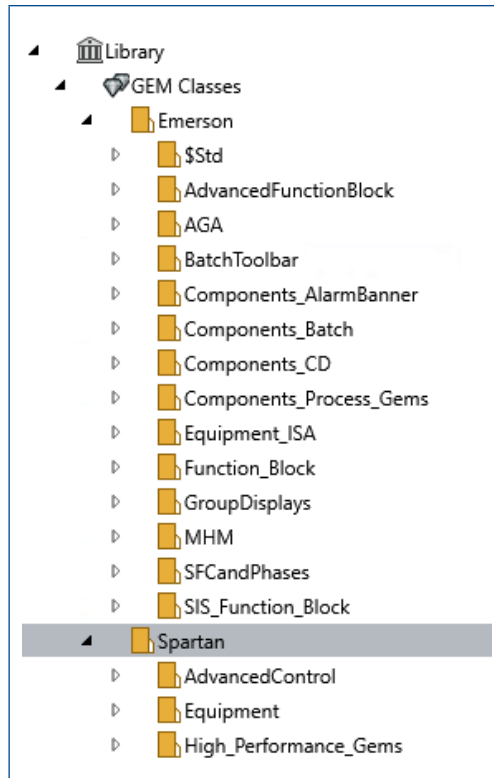
Figure 7. GEM Classes

Templates

The Template section is fixed with three groups of items: contextual displays, displays, and layouts.

Folder structures are not supported under Templates or within each category. For contextual displays, there are three Out-of-the-box templates provided. These are used when you create contextual displays for your project. You can create new templates with modified dimensions and default data.

Display templates can be created based on the type/size of displays you will want to build. The templates can have interactions and variables such that all displays created from a template will have those items pre-defined.
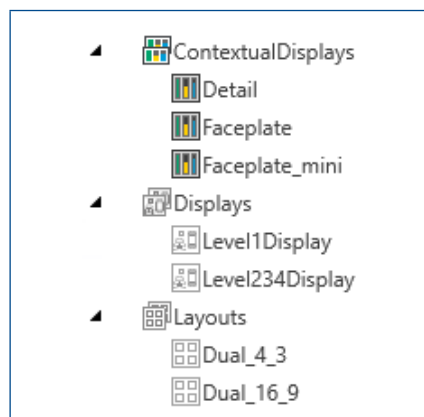
Figure 8. Templates section

The Layout templates provided include predefined variables used by some out-of-the-box functions. These variables act as global variables for the workstation and should be present on all custom Layouts created. Timers are also a feature of the layout, as well as interactions, such as "OnOpen", "OnClose", and "OnKeyPress", where scripts run independently from open displays and can initialize variables based on standard values.

Layout screens are defined based on the physical positioning of the monitors and can reflect their physical size. Intuitively, a 24-inch monitor can be defined in the layout to be 24 inches. Or you can use centimeters or points.

(1 inch = 2.54 cm = 72 points)

This is intended to allow the user to create screen objects that will appear with accurate dimensions. i.e. a 1-inch circle will measure 1 inch on the screen. However, there are several factors to consider when setting Screen size within the Layout.

First, the dimensions of the Layout Screen will define the size of the Menu Bar and the size of the navigation bars in the Display Frames. The Screen Menu Bar and Display Frame Navigation Bars have a fixed size. The larger the screen size in the layout, the smaller the relative size of the Menu Bar and Navigation Bar.

> **Tip:** If the Menu Bar and Navigation Bar are too small, redefine the layout by making the screen size smaller. Note that this will result in less vertical space on the screen for the display area and will result in the display area aspect ratio shifting slightly.

Secondly, the size of a display will dictate the relative size of GEMs as you place them in the display. If you make the Display size too large, the GEMS will appear too small, and vice versa. The Out of the Box GEMs and faceplate displays have been created with a 24-inch HD monitor in mind, with a single main picture Display Frame and an Alarm Banner Display Frame.

Thirdly, DeltaV Live does not require the Display sizes to match the Layout Display Frame dimensions. In fact, a display is likely to be used on different console arrangements where the Display Frames differ in size. The important dimension to consider between your Layouts and Displays is the aspect ratio. As you build layouts for different console arrangements and monitor sizes, keep the aspect ratio of the Display areas consistent. They need not be the same. At run time, DeltaV Live will present a display within a display frame and maximize the size until the vertical or horizontal dimension is reached. The display is centered in the frame and the unused portions are referred to as letter boxing. The default display background Color Standard should be defined in the layout. The result is only minor letter boxing on certain layouts that have a slightly different aspect ratio than the display.

When creating a display, Graphics Studio provides an option to set the display size to match a Layout Display Frame size. However, you may want the GEMs to appear larger or smaller within the Picture.

> **Tip:** Use the Match Display Frame option to set the initial aspect ratio of the display size to match the target display frame. Then, lock the Aspect ratio of the display and adjust one dimension to achieve the desired size of the GEMs in the picture.

Standards and Functions

The Standards and Functions containers in Graphics Studio provide a means to implement the HMI style guide that should be developed as part of the display specifications. The Style Guide defines how displays will look, including font type and size, colors, images, etc..

Standards are defined constants or references that are manually set in the library and can be defined for icon images, colors, fonts, numbers, and Boolean values.. Functions are dynamic and will return a value based on the its defined inputs and algorithm. The value returned by a function can be any of the data types in standards.

When a display property is defined, it can either be defined as a static value in the display, or it can be referenced to a standard or function. The browser will show only valid standards or functions based on the property's data type, e.g., If defining a color property, only color standards or functions will be displayed.

The following display property is configured to use a standard for display background color, which is located in Library/Standards/Emerson/Common folder, but the reference is defined as:

Figure 9. Standard assigned to a color property

The actual color is defined by this standard and is also defined for each theme applied in the system. By using this standard in all displays, display frames, etc., a single adjustment to the system allows all display and layouts to respond consistently.
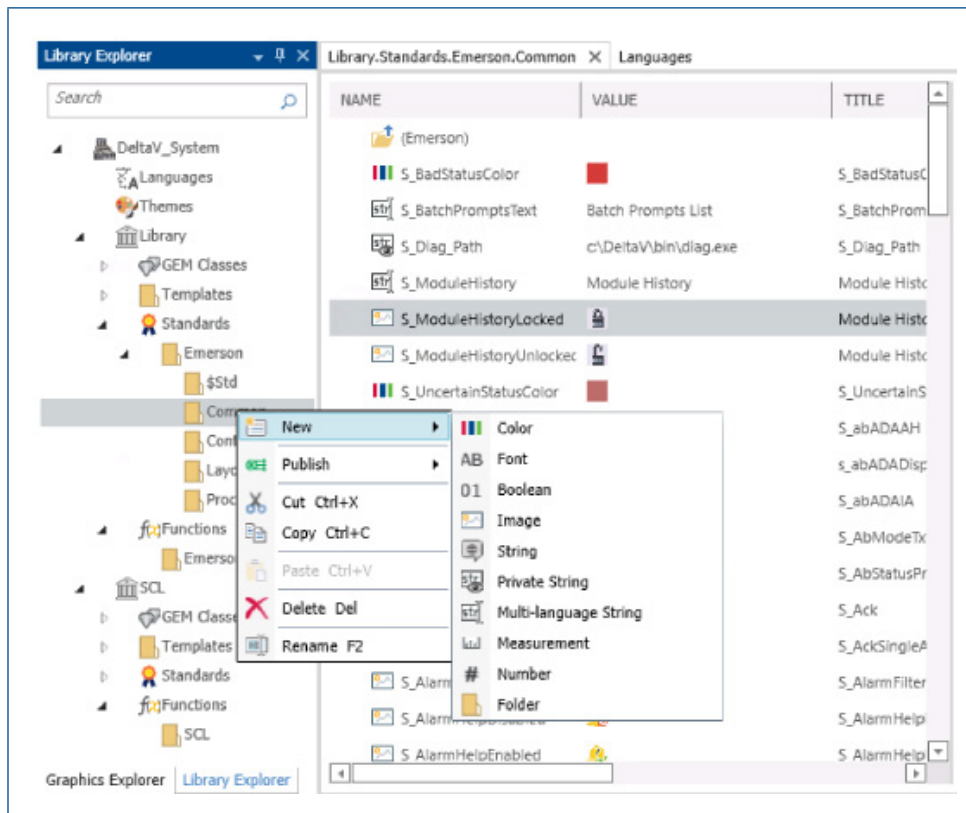

Figure 10. Library standards

There are nine standard types, each defining a property type. You may discover the need for new standards as the project evolves. Existing standards should be reviewed to ensure one does not already exist for the new application. The MOC for the HMI specification and Style Guide should be followed to avoid a proliferation of Standards in the database that are not defined in the style guide.

There are seven types of functions, based on the data type they return. Within a function you define the inputs, calculations, and conversion table that define the value to be returned to the property. Placeholders for the inputs are created in the function definition but are bound to a source at run time to show actual data values. Calculations can be added to manipulate and condition the inputs. The result is defined in the conversion table, which can be rule-based, value-based or script.

For instance, if you want to dynamically change the width of a line based on some logic, you need to return a measurement value for the line width property. The property in the display object has the option to be animated as indicated by the icon. The drop-down menu allows you to select a function, and the list will show you existing functions that will return a "measurement". You can choose to create a new function if one does not already exist that meets your needs.
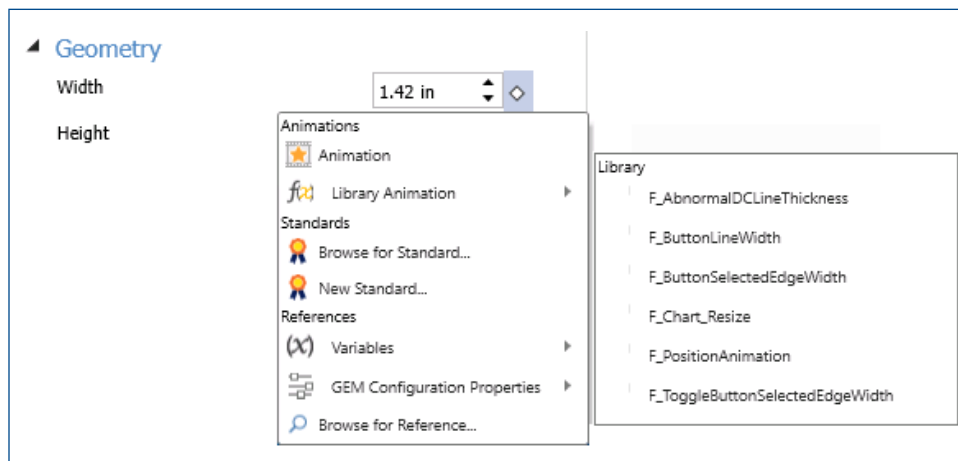

Figure 11. Defining a geometric property with a function

Is a Separate Library Needed?

The question comes up as to when should a new library be created? DeltaV Live supports multiple libraries in the Library Explorer, but you should first ask yourself if adding a library will make it easier or harder to manage your project. If objects are created using out-of-the-box (OOB) items, they will contain links to the OOB library for those items, creating dependencies across those libraries. Libraries are independent Name spaces. Within each Library, names must be unique. Items copied across libraries can keep their original names. Note that Library items are referenced by Library.Item_Name.

One application where a new library might serve you better is if you are integrating an OEM subsystem that is provided with a set of custom display objects in a dedicated library. The OEM may maintain support for this library as part of their scope, and it will be easier to maintain this library as a separate subsystem in the database.

As an alternative to a new library, you could create additional folders in the OOB library instead. In the example below, the library has been augmented with site-specific folders for the company ACME. Any new items can be added to these folders, and this helps keep the OOB items in their original form.
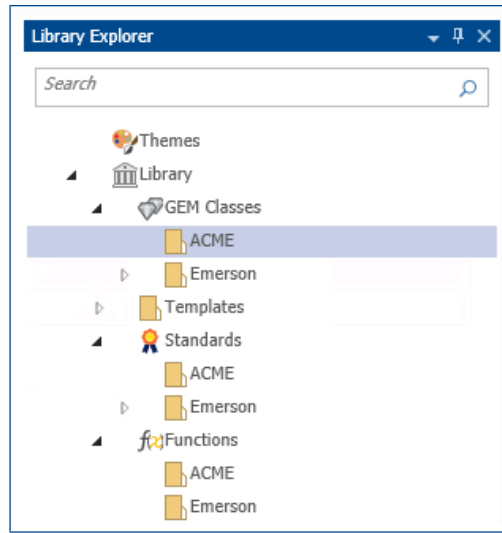
With this approach, the user will add any new items in their site folders. They can create new items by copying existing OOB items as a starting point and renaming them appropriately. They can also choose to move any OOB items to a site folder so that their site folders contain a complete set of library objects to be used in the project.

Moving items between folders within the same database without renaming them does not affect any existing links or references to those items.

You can also choose to rename the Emerson folders and add your project specific items in the existing folder tree. However, this makes it more difficult to discern OOB items from modified or added items. In this scenario, it may be best to use a site naming convention and, whenever an OOB item is modified, rename it to your site standard prior to modifying it. In this way, you can easily find site specific items vs OOB items.

> *Tip: Keep things simple. Create folders and leave unmodified OOB items with their original names.*
>
> *If you want to modify OOB items, you can move them to different folders within the same library and rename them, and the database will manage the changes in all existing references.*
>
> *If you simply copy items to a new library, renaming them in the new library **does not** update the existing references.*

To rename an item, open it and view the Graphics Configuration pane at the upper level of the item. Select the name in the header and rename it. Close and save it. If this is a heavily used item, it can take a minute or more for the database to completely update all references. Publish any items affected by the change.
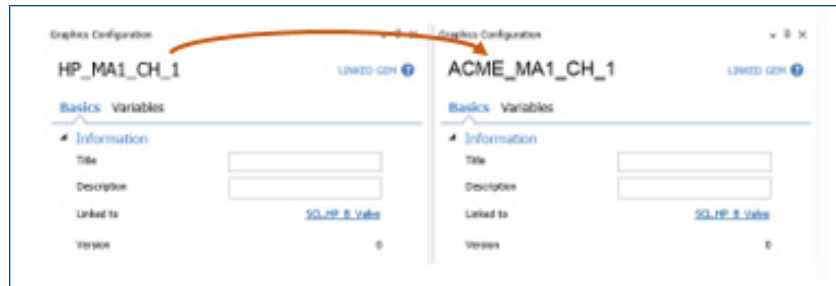
Figure 12. Graphics Configuration Pane

Do not forget to back up your database. Once you've defined your library and site-specific items, make sure you export and back up the database before you proceed to building displays.

Setting up Graphics Explorer

The Graphics Explorer is where you will create and organize your project displays, display sets, contextual displays and layouts. The Displays folder comes with some OOB displays located in the Emerson folder. These focus on the alarm banner, alarm displays, and some batch-related pictures.

Displays

The first step will be to setup a folder tree for your displays. The explorer tree can extend to 4 folders deep. Keep in mind that the ISA 101.01 standard calls for Level 1 (L1), Level 2 (L2), Level 3 (L3) and Level 4 (L4) displays as well as "Other" displays. You could set up the folders to reflect your display hierarchy, but in reality, the Display Sets will define the Hierarchy. There is no need to do this twice.

It is recommended to create a Folder structure that mirrors the DeltaV Plant Area definitions. Each folder contains the L2 and associated L3 unit displays. Each display can be assigned to an alarm area or unit, which then serves to display the active alarms per display. The L1 (Overview) displays can be placed in their own root folder and represent the various spans of control for different operators. Alarm rollup on the overview is based on the logged-on user.
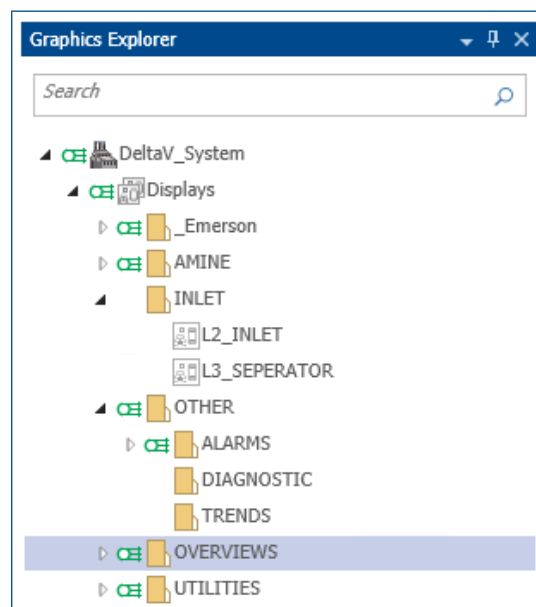
In addition to the hierarchical displays, each display set contains a list of non-hierarchical displays, such as the alarm lists, alarm suppression, alarm filter, DeltaV Help, network diagnostic or utility displays. Create a folder for these display types, with sub-folders to help manage them.

> *Tip:* Create a meaningful structure with a limited number of displays per folder and avoid a long flat list that requires a lot of scrolling.

The displays in the hierarchy are all intended to appear in defined display frames. As such, their size and aspect ratio should be consistent throughout. Some workstations are used for showing central data such as KPIs. They may have custom size settings and dedicated displays. You can decide where to place these in the display tree, and they would work with a custom display et and layout for those types of stations.

Display Sets

As you develop displays, you will also assign them to display sets to define the navigational hierarchy. The hierarchy follows the ISA 101.01 standard, with an L1 Overview containing L2 Area displays, and each Area display containing L3 Unit displays, and finally, each Unit may also have L4 equipment displays. This mirrors the ISA 95 model for asset hierarchy. You may choose to use only L1 and L2 and some L3 displays. You can also have multiple L1 displays in a single Display set. The position of displays in the hierarchy defines the Up, Down, next and previous displays for scrolling through the displays.

The display sets define which displays are available at each workstation. During workstation configuration, multiple display sets can be assigned as needed to serve the intended purpose of the workstation. The active display set defines the working hierarchy for the operator and the DeltaV security defines his span of control. Each Operator's workstation should have a Display Set that mirrors their assigned Plant Area, and includes all the L2 Area Displays that align with their assigned Plant Areas.

In addition to the hierarchical displays, non-hierarchical displays added to the display set to round out the user experience. Checking the option of "Include all displays"" means all displays will be published to the workstation. As you refine the operator environment, you can develop a prescriptive list of displays. Until this is defined, you can keep the "Include all displays" option selected. Remember that user security will prevent an operator from acting on control modules not assigned to their span of control.

If display sets and hierarchy are not part of your site's HMI specification, consider adding a section to help in the implementation of this feature.

Contextual Displays

Contextual displays include faceplates and detail Displays. They are designed for general use and are assigned a context when they are opened. Each DeltaV control module is assigned three displays: primary process graphic, detail graphic and faceplate graphic. The display named in the control module configuration determines the contextual display that is called. If you are migrating a DeltaV system, there must be a matching contextual display for every detail or faceplate display that is referenced in your modules.

Like the DeltaV Operate displays, contextual displays are dynamically linked to the context module when they are opened. Each display can serve all control modules that share the same control information, such as PID1 block.

The OOB contextual displays that have been updated in DeltaV Live have a new look and feel. You can also create your own displays with a converted look and feel that is consistent with DeltaV Operate. Consider your site's HMI specification and whether you should update this document to use new display designs rather than recreate the same old designs to match the specification. Migrating to DeltaV Live is an opportunity to improve on your designs.

Layouts

Layouts provide a mechanism to define the functional areas of the operator console. A console is made up of one or more monitors arranged to serve the needs of the operator. Within each monitor, displays are used to represent process information in a consistent fashion, including alarm system data and historical trending.

Layouts are made up of screens and display frames. A screen can be any size. One of the first things to understand about DeltaV Live is that monitor resolution does not matter. Displays will be accurately presented on any screen resolution because displays are HTML5 graphics using scalable vector graphic elements and will properly fit displays within their available area. It is important to understand that the Screen Toolbar and Display Frame Navigation bar have a fixed dimension. The larger the Screen size defined, the smaller these will appear relative to the screen size. In reality, they will be the same size on any monitor if the Screen Size matches the physical size of the monitor.

> ***Tip:*** *If you want the Menu and Navigation bars to appear bigger on a given monitor, define the screen size to be smaller than the physical size of the monitor.*
>
> *However, changing the screen size means redefining the layouts, so it may be worthwhile to resize the Menu and Navigation bars first for the various monitor arrangements.*

Figure 13. Layout Configuration showing two display frames

The Menu Bar can be described in three sections: buttons that are not configurable, buttons that can be configured, and buttons that appear dynamically. The image below shows these sections in runtime. For the buttons that are configurable, users can choose which ones are enabled or not from the Graphics Configuration properties. Some buttons are tied to specific displays and others to DeltaV applications.
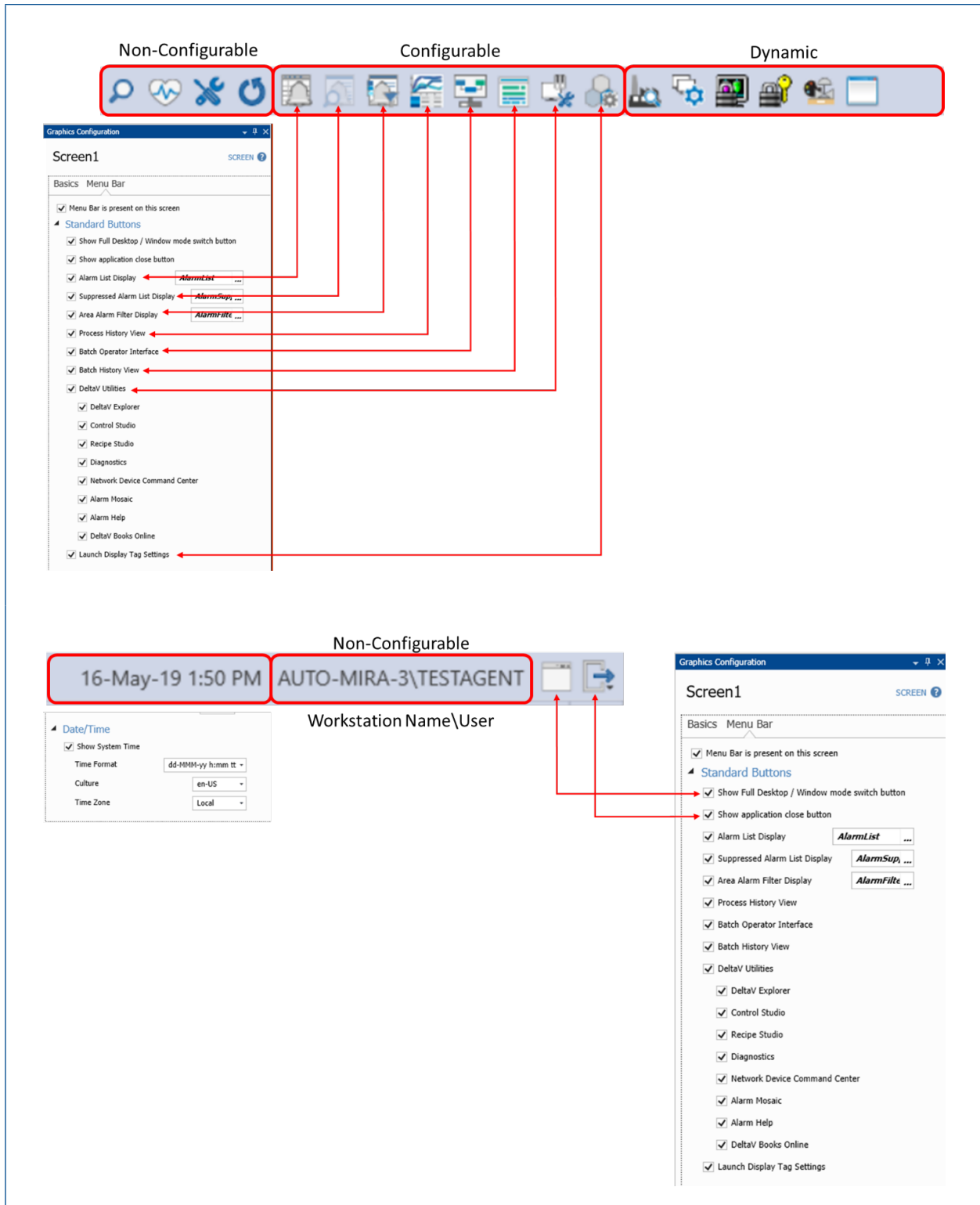
Figure 14. Menu Bar components

The dynamic section in the Menu Bar will display buttons for other applications open on the workstation and will be visible when DeltaV Live is running in full screen mode, eliminating the need for the "Alt-Tab" application list. Also, different buttons will appear when you minimize a popup display, a faceplate display, a detail display or watch areas. Multiple items of the same type will be grouped under the same button in the Menu Bar (e.g., there will be one button for minimized faceplates and another for minimized watch areas).

In addition, users can add up to 5 custom displays to the Menu Bar for easy access to the Operators. When using this feature, displays must be assigned in order starting with the Custom Button 1.
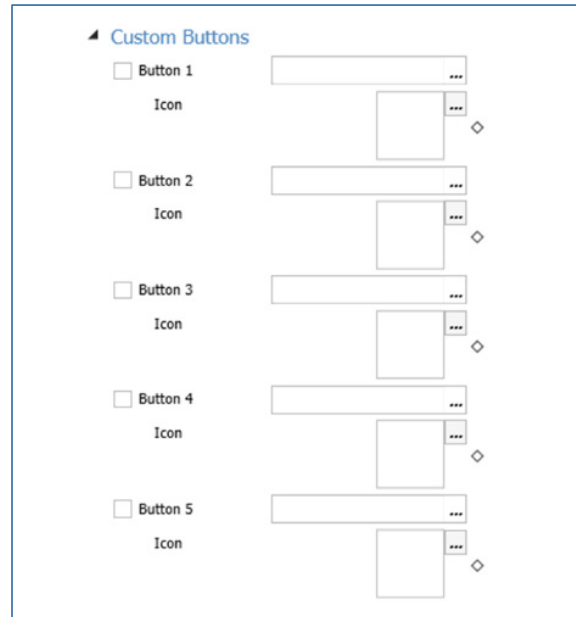

Figure 15. Custom display buttons for the Menu Bar

The second part of the layout is the display frame. The example in Figure 13, shows two display frames: one for process displays, equipped with a navigation bar, and a smaller alarm banner display frame, which will have a fixed, or static display. The Navigation Bar is given a starting level in the hierarchy and how many sublevels to show, and it can be adjusted to show 1, 2, or 2 rows. Alternatively, you can define the navigation for the frame to be a flat list of all assigned displays on the workstation.

In a single monitor workstation, any and all displays would appear in this frame. On a dual or quad monitor workstation, certain display frames can be dedicated to only show a specific level of display, like L1, L2 and/or L3 displays. This complies with the ISA 101.01 standard. In each display frame, you may choose to use a single row, dual row or three row navigation bar. Finally, Automatic Display Coordination can be enabled such that upper level displays will change based on selection of lower level displays. Opening an L3 display from a different area would cause the L2 display for that area to open.

In addition to the navigation bar, traditional display link buttons in process graphics allow navigation within displays. Also, objects can be associated with out-of-the-box actions to navigate to displays, maintaining typical navigation mechanisms and making it simple for engineers to configure multiple ways to achieve the different philosophies around this topic.

Display size vs. Monitor Size

With DeltaV Live, you can create displays of any size. These will be shown on consoles with monitors of various sizes defined by layouts that may or may not physically match the monitor sizes. How do you decide what size to use for displays?

As we explained earlier, layout screen size dimensions dictate how big the Menu Bar and navigation bar will appear on a given physical monitor. Within a layout screen, you will define display frames with an overall size. The Navigation Bar rows will then determine the size of the effective display area in each display frame. How big this area is in points (pt), inches (in), or centimeters (cm) does not really matter for your displays. What matters is the aspect ratio of this area.

It is important to make displays match the aspect ratio of the actual monitor. How big we make the displays will impact the relative size of the GEMs.. So, the size of your displays will be influenced by the GEM library.

DeltaV Live will maximize the display area used to show a display without distortion. Displays that have a different aspect ratio will work in any display frame, and any unused area will be letter-boxed. Therefore, to maximize your display area, keep all layout display areas relatively close in display area aspect ratio. Then, your displays will appear with minimal or no letterboxing.

> *Tip:* Here are some rules of thumb for layout file and display size settings:
>
> 1. *Set your layout screen size to provide the desired Menu Bar and Navigation Bar sizes for the use the screen is intended for. It is not possible to set the Menu Bar size explicitly, so the Layout's screen size must be compromised to achieve the desired presentation.*
>
> 2. *Set your display size to reflect the desired presentation of your dynamos. Do not worry about the layout's display frame size, except to match the aspect ratio.*
>
> 3. *Pay attention to the display frame sizes in different layouts and try to keep aspect ratios consistent across your layouts. But don't worry about a few points (pts) here and there. The displays will letterbox and appear as intended without distortion.*
>
> 4. *Adjust your alarm banner height after you have defined your desired display frame display area. You may want to have different alarm banners to fit into the alarm banner display frame of different layouts to maximize the display frame areas.*

When it comes to displays and layouts, everything is relative. It does not really matter what the size of a monitor is or the screen of the display. It is the interaction of the GEMs in displays and displays in layouts. You will need to evaluate the GEM library items to determine display size, and each layout will be defined by the application and monitors.

It is important to remember these rules if your site's HMI specification calls for specific items to be a certain size, as Graphic Studio allows you to define objects to these measurements.

Initial Displays and Layouts

While configuring layouts, you can define initial displays for each display frame that will be shown when that display frame launches or when the layout is reset. The frame type can be dynamic or static. If the display frame type is set to "static", you will not be able to switch its content to a different display. This is commonly used for alarm banners or toolbar displays. If the display frame type is set to "dynamic", which is the setting typically selected for process displays, you can switch its contents.

DeltaV Live supports two alternatives to configure initial displays: Named Display and Position in Hierarchy. When a Named Display is selected, the configuration interface allows you to browse and select a display to assign to the display frame. A common example could be the selection of an overview display that will always open in the top left screen of a quad monitor layout.

When Position in Hierarchy is selected, you can define displays without specifically identifying them by name, but rather make the display selection relative to a position in the active hierarchy, this implies that a display set has already been created and defines this hierarchy (refer to section Display Sets). For example, on a given display frame, the frame will resolve the initial display according to the position in the active display set for that workstation.
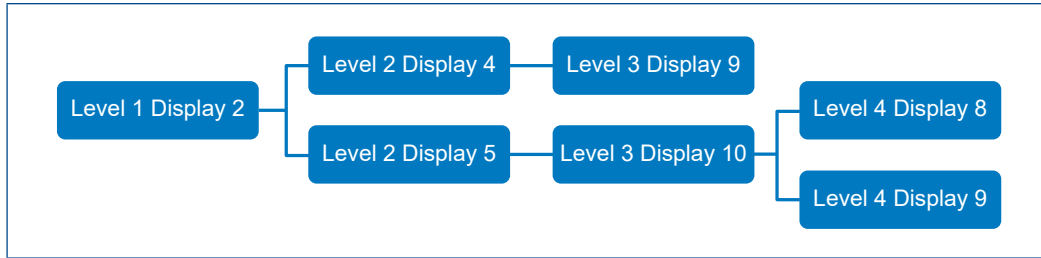

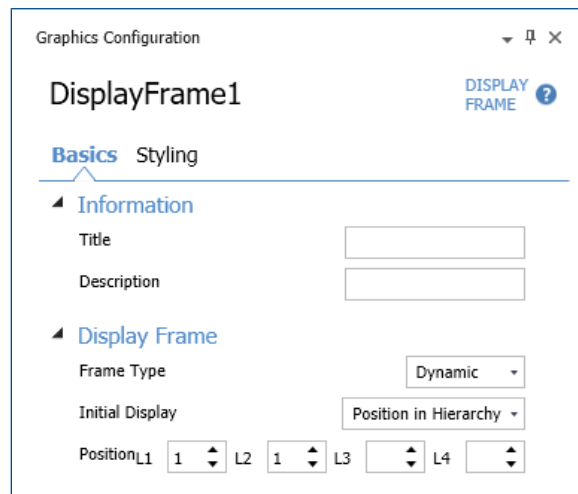*Figure 16. Display Set 1*

Figure 17. Display Set 2

Figure 18. Dynamic assignment of initial displays by position in hierarchy

Since there could be multiple display sets assigned to a workstation, selecting Position in Hierarchy allows for relative allocation of initial displays and allows for reusing the same layout on multiple workstations. For example, when using Display Set 1 and 2 from above images, the same Layout renders different initial displays in workstation:

Workstation	Display Set	Layout			Initial Display
		Display Frame Type	**Initial Display**	**Position**	
Workstation 1	Display Set 1	Dynamic	Position in Hierarchy	L1: 1 L2: 1	Level 2 Display 1
Workstation 2	Display Set 2	Dynamic	Position in Hierarchy	L1: 1 L2: 1	Level 2 Display 4

The advantage of using Position in Hierarchy is that switching display sets will also automatically switch initial displays for each display frame. For a given layout, the desired initial displays across all display sets must be in the same hierarchy positions. This allows you to use the same layout across different workstations and simply assign different display sets to the workstations to change which display is designated as the initial display.
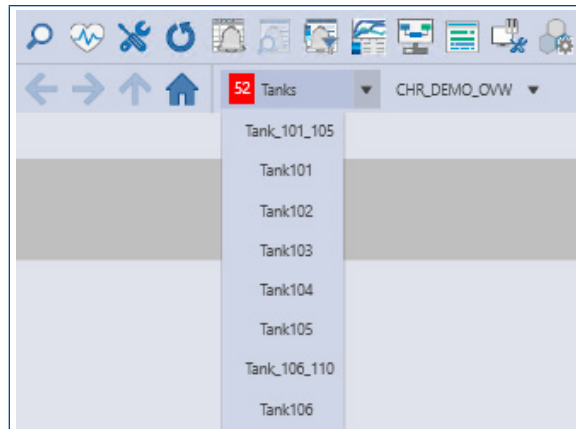
Figure 19. Alarm rollup on the Navigation Bar

Alarm Rollups

Alarm rollups are an option that can be added to the Navigation Bar. An alarm rollup shows all active alarms from modules, devices, and nodes from the parent display and children displays.

Alarm rollups can be configured as one of three selections:

1. No rollup

2. Primary control display – counts alarms associate with the primary control display

3. Plant Hierarchy – counts alarms associated with the configured Plant Hierarchy for that display (area, unit, or equipment module)
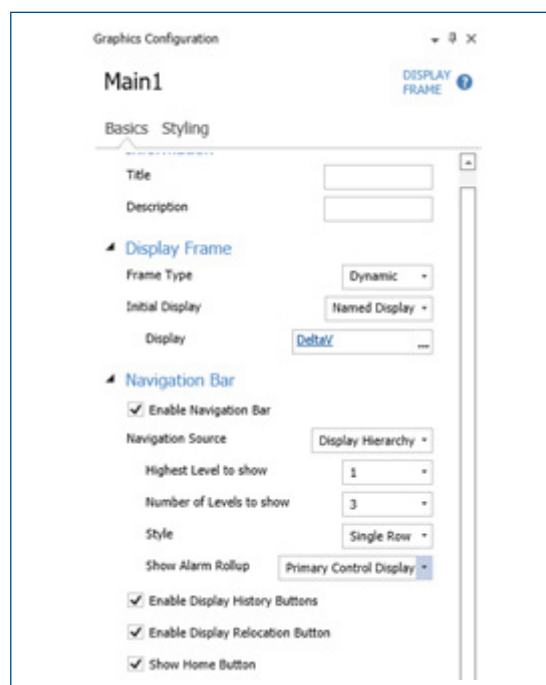

Figure 20. Alarm rollup options in Graphics Studio

Working with GEMs

DeltaV Graphics Studio introduces the concept of GEM classes. Although a GEM is similar to a Dynamo in DeltaV Operate, they differ in significant ways. GEMs are the building blocks that you will use for your displays and display objects. GEM stands for Graphical Element. You can incorporate GEMs into other more complex GEMs, allowing for common structures to be created and reused throughout the library.

Your HMI specification may have a variety of defined display objects, including a description of their animation or dynamic behavior at run time. Common items like icons or data groupings, alarm information, abnormal status, etc. can likely be broken down into basic common GEMs, and then assembled together in a purpose-built GEM. A review of the out-of-the-box GEMs will reveal how this can be accomplished. In combination with defined standards and functions from the library, the GEM library can be developed to satisfy the most demanding specification.

GEM Properties

When a GEM instance is created in a display, the Graphics Configuration pane reveals the various properties of that GEM. (This replaces the Dynamo Configuration dialog in DeltaV Operate). In addition to the default properties of display objects like name, geometry, visibility and such, there are typically GEM specific properties to be defined, such as a module name or orientation.
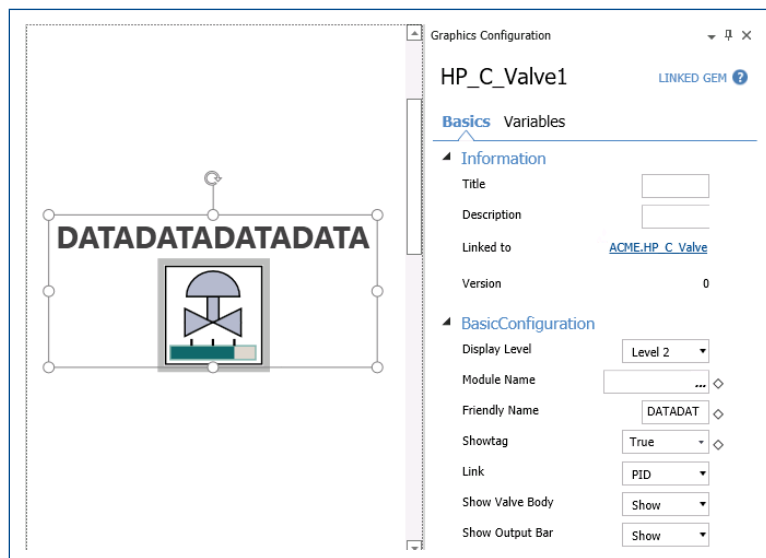

Figure 21. GEM instance properties example

In the example above, the control valve GEM is shown with its properties. The "Information" section is common to all GEMs and includes the "Linked to" property, which identifies the source GEM class, in this instance ACME.HP_C_VALVE. Selecting this link takes you directly to the GEM class where changes can be made to the class if needed. It is also possible to unlink the GEM and modify the instance to create a unique implementation independent from the class definition.

For more details, please refer to the DeltaV Live Working with GEMs whitepaper.

Building your Display Project

Now that the library has been defined to align with the HMI specification, the tools are in place to start building process displays, contextual displays, display sets, and layouts that will meet each workstation's functional requirements. As mentioned earlier, display design is likely an iterative process in the early stages where you may need to create new library items such as additional display templates or Contextual Display templates.

In Graphics Explorer, you will build items in their respective locations.. Unlike DeltaV Operate, you can create a meaningful folder tree in which to locate your displays, display sets, contextual displays, and layouts, These folders simply help keep things organized and easier to find.

Speaking of finding things, Graphic Studio includes a powerful search tool with partial search results that can quickly find any item. When you select an item and clear the search bar with the X button, the Graphics Explorer will return to the tree view with the item highlighted. You can also open the item from the search list by double clicking it.

The following sections will provide guided steps to create a display, starting with basic actions and slowly adding complexity to the graphic. This guide assumes that there is already at least one Control Studio module that can be used for displaying parameters or creating animations.

Building a Simple Display

To begin working on a display, it is necessary to create the empty display by:

1. If Graphics Studio is not already running, launch it.

2 From the ribbon, select New, Display, Blank Display. "Display1" is created. Notice that Graphics Studio includes two templates that have basic configuration ready to use; feel free to explore those templates as well.
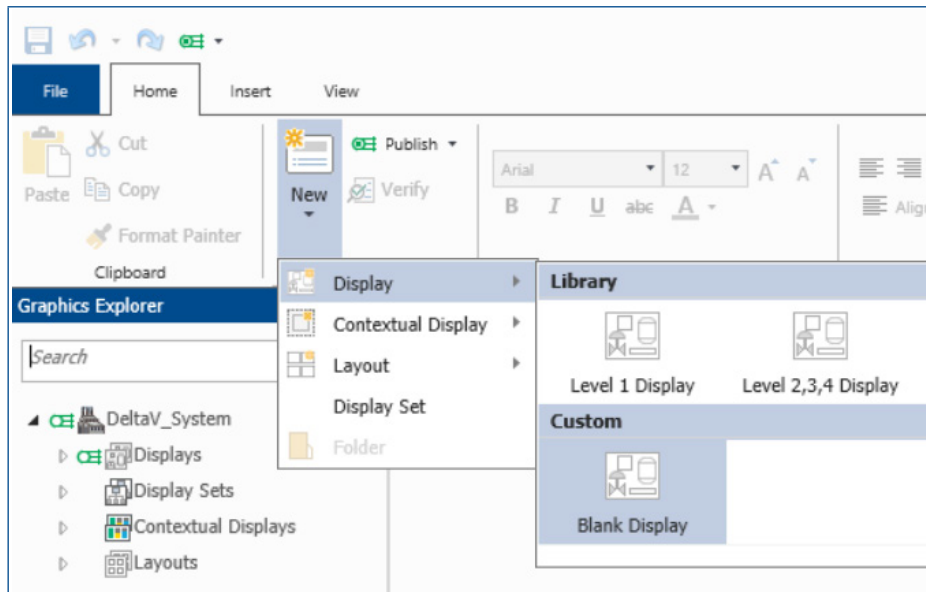

Figure 22. Creating a new blank display

3. From the Drawing section of the ribbon, select a rectangle shape, then click and drag over the display area to add a rectangle to the display. "Rectangle1" is created.

4. With Rectangle1 selected (see white adorner circles around the object), observe the Graphics Configuration pane to the right of the display canvas is populated with Rectangle1 properties. Explore changing the rectangle properties such as Fill Color, Fill Percent, Line Color, and Line Thickness.

5. From the ribbon, select the Line shape, then click on the display to add a line to the display by clicking on a starting point and dragging to the ending point, releasing the mouse will place the line on the canvas.

> **Tip:** *Pressing the SHIFT key will force the line to snap at 45° increments, according to the mouse direction.*

6. Select a line endpoint (white adorner circle) with the mouse and observe the endpoint can be moved to a new location. Move the line endpoint over Rectangle1 and observe 4 red "connection points" appear. Attach the line endpoint to a red connection point.
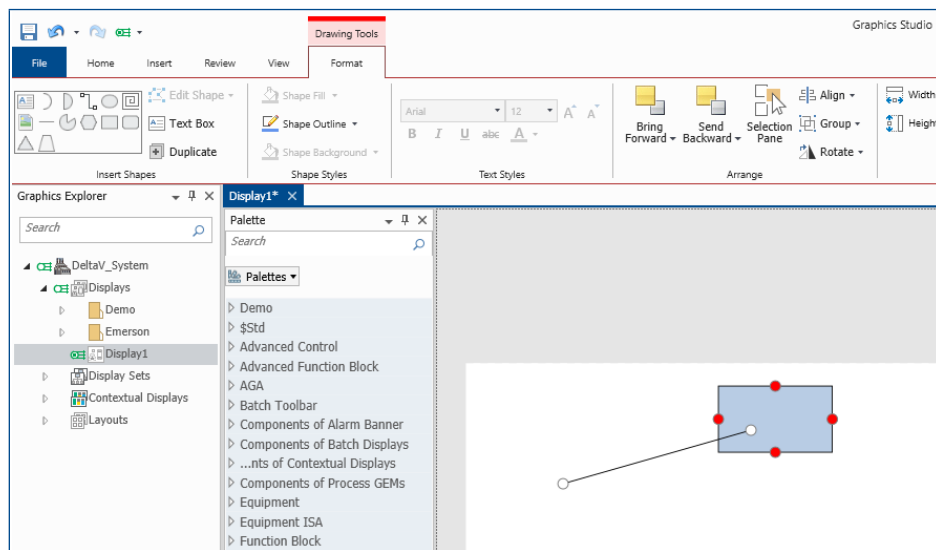

Figure 23. Red adorner circles are connection points

7. Select the rectangle and move it around the display. Observe the line endpoint remains connected to the rectangle.

8. Create a new horizontal line, "Line2". On the Graphics Configuration pane, observe Line2 has several properties: Color, Thickness, Style, Arrow Start Type, Arrow Start Size, Arrow End Type, Arrow End Size and Crossover Effect. Explore setting different values for these properties, for example: Arrow End Type = Normal, Arrow End Size = Medium.

9. Create a new vertical line, "Line3". Set the Crossover Effect property for Line3 to Gap. The crossover effect is designed to automatically detect where lines intersect and show that intersection as a gap or jumper on the line that has the property enabled.

10. Take the endpoint of Line3 and move it so that Line3 intersects with Line2. Observe the automatic Gap that is created.
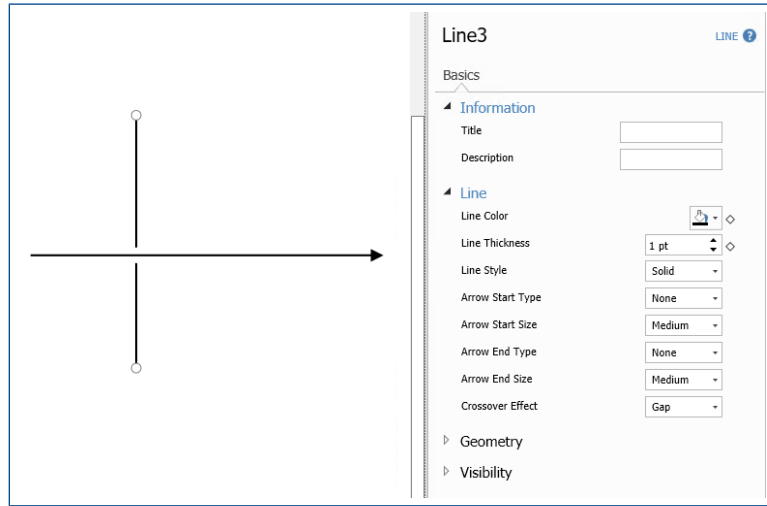

Figure 24. Crossover Effect Gap

11. From the ribbon, select the connector shape, then click on the display to add a connector to the display. Observe the connector shape has similar properties to lines, except the connector has built-in 90-degree angles. Depending on the connector's routing, its segments can be adjusted by moving available yellow handles.
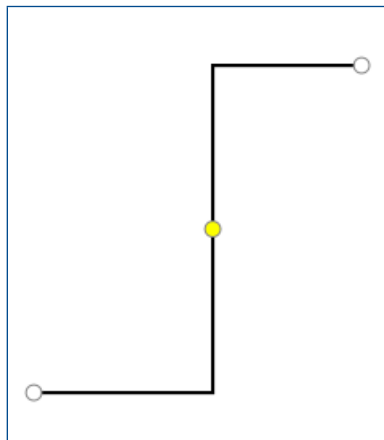

Figure 25. Connector shape

12. Observe the Display1 tab has an asterisk next to the name. This means there are changes pending to be saved. Click Save in the upper left corner of the application using the disk icon or use CTRL + S to save using typical keyboard combination.

Figure 26. Indication that changes are pending to be saved

Welcome to DeltaV Live

Adding a datalink

1. One of the most common elements in a display is a datalink. These objects display data from DeltaV by connecting the display object to a DeltaV parameter. To add a datalink, find it in the Palette and drag it onto the canvas. It is possible to search for it in the Palette search box or expand the Data section of the palette.
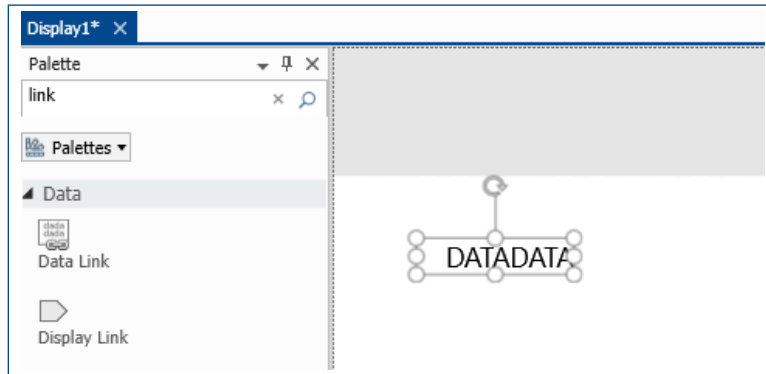
Figure 27. Palette search field and datalink on canvas

2. In the datalink configuration pane Basics tab, find the Expression field (under Data Source) to browse for DeltaV control parameters. Click on the ellipsis button ( ) to open the parameter browse and navigate to the parameter needed.
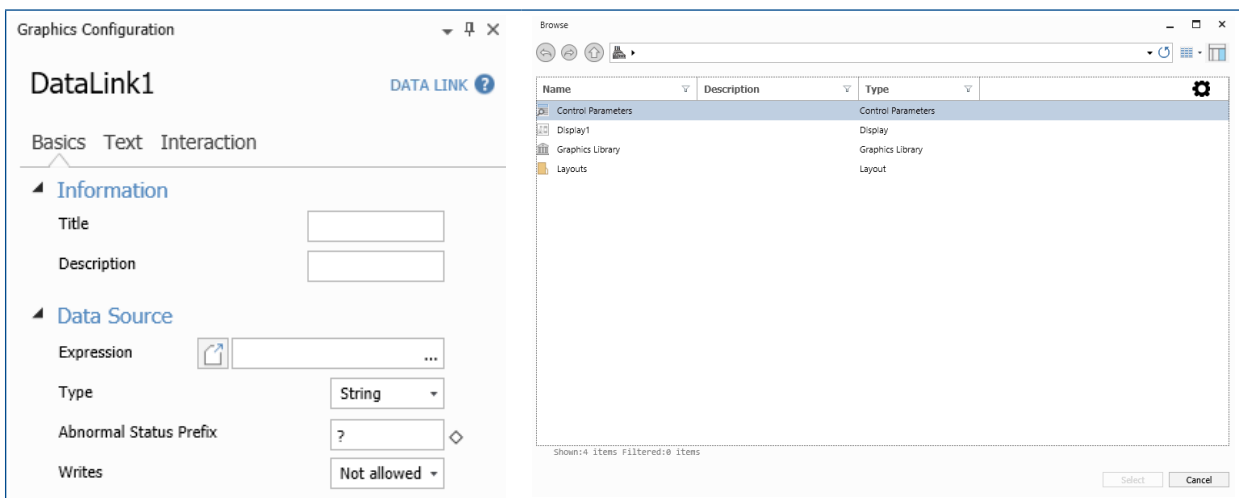
Figure 28. Data Link Configuration Pane and Parameter Browser

3. Once selected, the DeltaV parameter will be shown in the Expression field. Hitting the "Enter" key or tabbing away for the field will verify the validity of the path and indicate to users if it can be resolved or not. When a path is properly resolved, it will be indicated with blue color and solid underline. When the path can't be resolved, it will be indicated with red color and squiggly underline. There could be two reasons for an unresolved path: there could be a typo in the cases where the path was typed in instead of browsing to it; or it could be that the module is not present in the DeltaV configuration database, possibly because the display was imported, and the control module was not. In the image below, notice that the same path has been broken by introducing a typo.
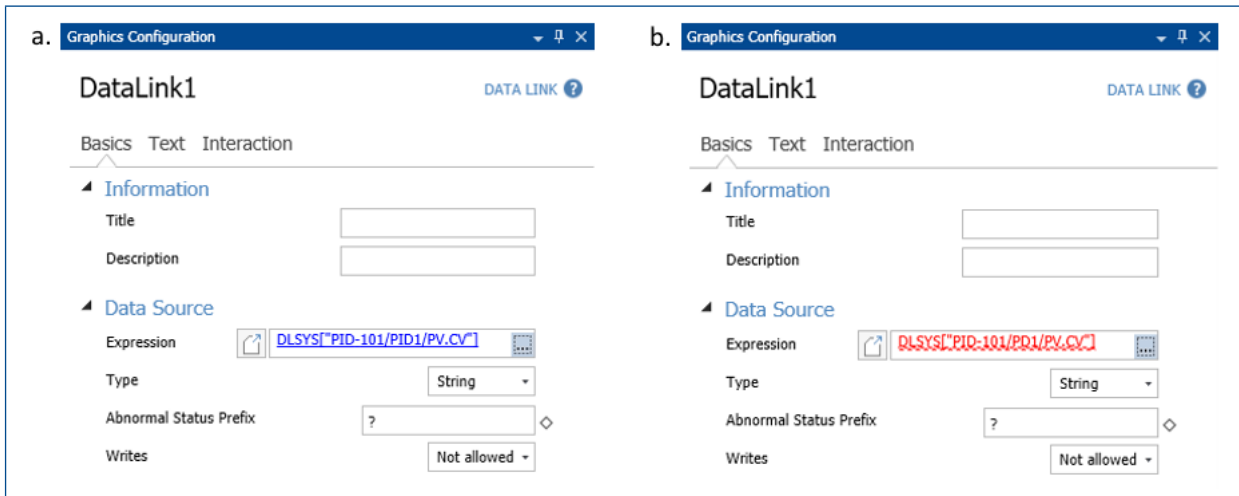

Figure 29. Resolved (a) and unresolved (b) expression for a datalink

4. Change the Type to Numeric and choose the number of decimal places needed. The number of places to the left of the decimal point is determined by the width of the datalink object.
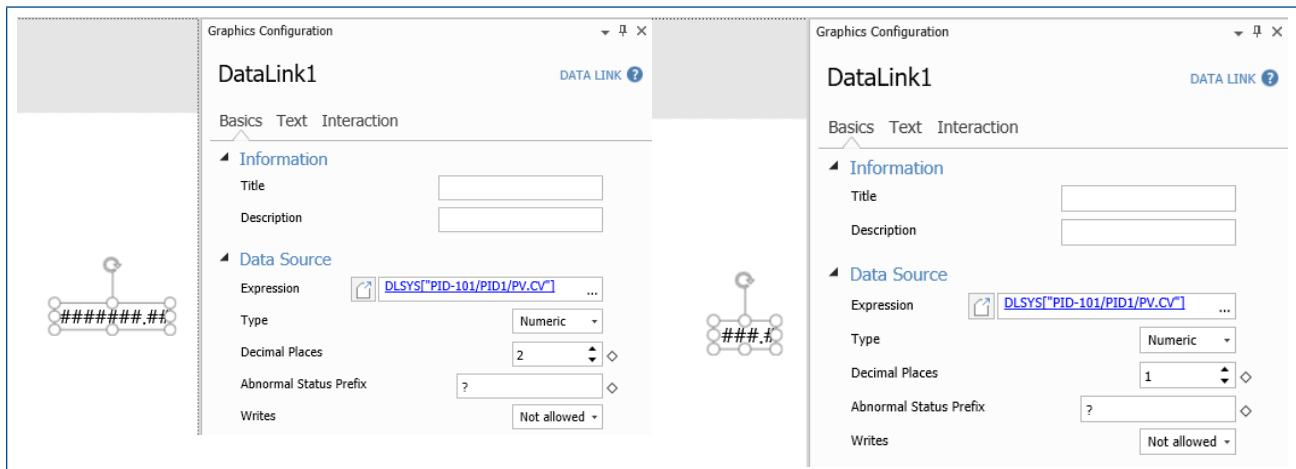

Figure 30. Changing Whole and Decimal Places in a Datalink

Adding a level animation

1. Another very common element in a display is an indication of level. In Graphics Studio, any shape can be animated to represent a dynamically changing level. For this example, a Rounded Rectangle will be used.

2. From the ribbon, click on the Rounded Rectangle shape and drag to draw it on the canvas. Adjust the yellow adorners to provide the shape with the desired form.
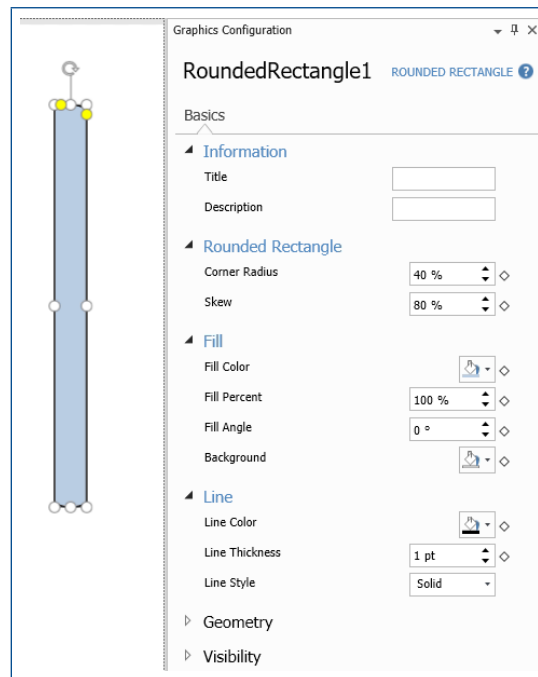

Figure 31. Rounded Rectangle Properties

3. The property Fill Percent, under Fill, allows for animating the filling of the shape. Click on the diamond and select Animation.
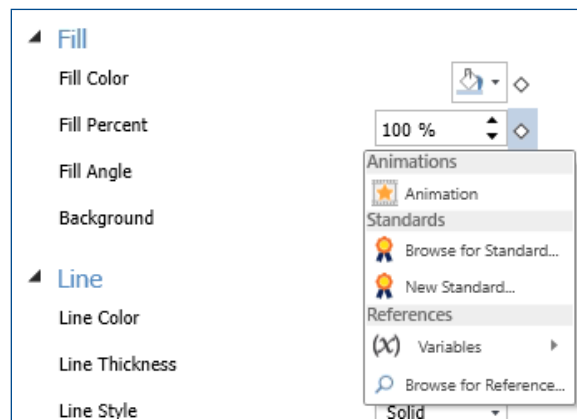

Figure 32. Enabling Fill Animation

4. The Configuration Pane updates to display all required fields to complete a fill animation configuration.
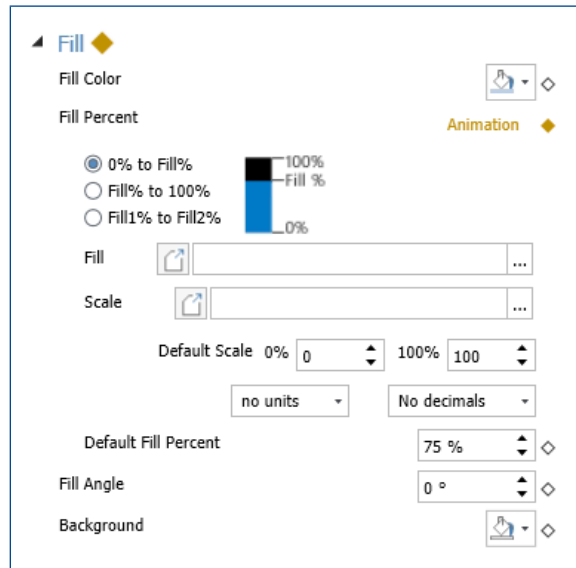

Figure 33. Fill Animation Properties

5. For this example, the default 0% to Fill% animation will be configured, but feel free to experiment with the other two options.

6. If the Fill Parameter is associated with a Scale Parameter, browse for it from the Scale property. DeltaV Live will assign the scaling according to the scale parameter provided and will ignore the Default Scale properties. When the Fill Parameter is not associated with a Scale Parameter, enter the Default Scale information to map the bottom and the top of the shape to the actual bottom and top values for the fill animation.
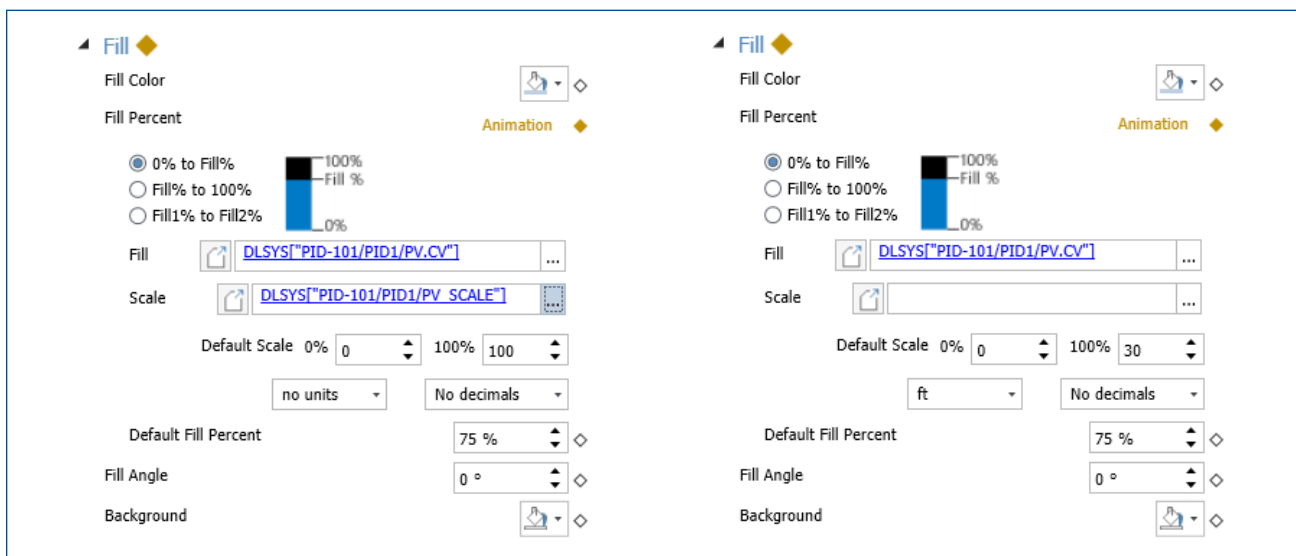

Figure 34. a) Fill animation with scale parameter. b) Fill animation with default scale

Saving and publishing changes

After saving, notice that the green publish icon next to Display1 on the Graphics Explorer pane momentarily disappears and comes back. Publish the display by clicking Publish on the ribbon. Publishing can be done from the ribbon, in context from the object, or from the Quick Access bar (top of the screen), where it has been placed for convenient access.

The green icon with filled center indicates that a configuration item has publishable changes pending, and the green icon with clear center indicates that a container or folder contains at least one configuration item with publishable changes pending (see Figure 31 below). When a configuration item has been added or modified, a publish pending indicator appears next to its name in the Explorer view, or for workstations, in the Workstation Management dialog. When a GEM class is modified, the displays containing its linked GEMs, rather than the GEM class, show a publish pending indicator.
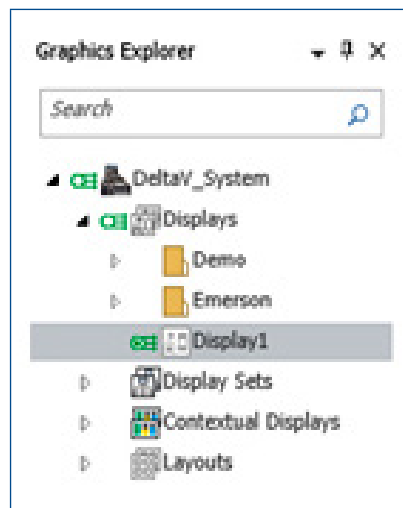

Figure 35. Publish icon

> *Tip: The publish indication bubbles up to the top of the hierarchy to facilitate identifying where are the objects pending to be published.*

Publishing authorizes graphics configurations to be deployed for use online. When a graphic configuration is published, it is marked as available for a workstation to retrieve it. Operators receive an indication that updates have arrived, but to ensure that publishing does not interrupt their normal workflow, they choose when to accept those updates by clicking the refresh button in DeltaV Live.

Figure 36. Runtime indication that changes are pending

Welcome to DeltaV Live

When the Publish command is selected, the Publish dialog opens. The check boxes are automatically selected next to all items that are eligible for publishing. If any of the selected items are not ready to be published, deselect the check box next to it.
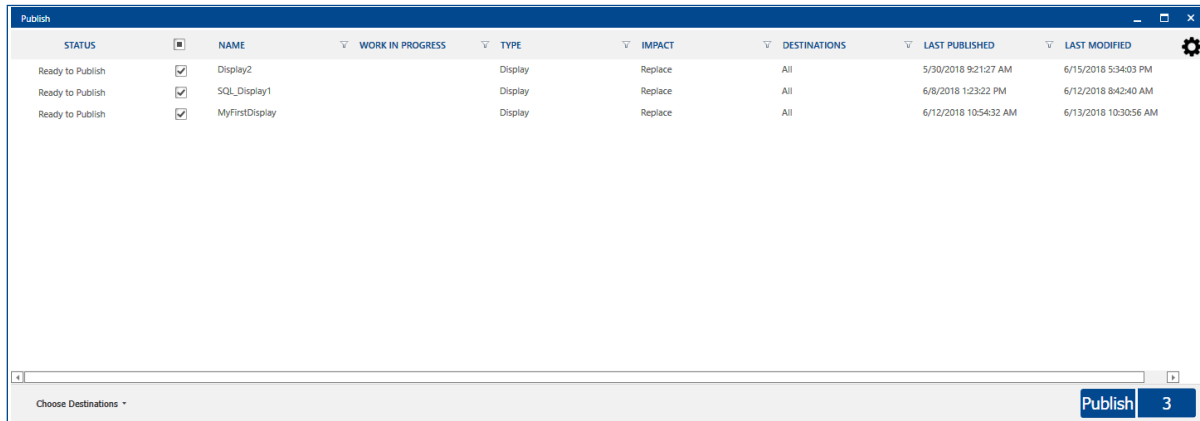

Figure 37. Publish dialog

> *It is possible to publish to a specific workstation by selecting it from the Choose Destinations option in the bottom left corner of the Publish dialog.*

Un-Publishing configurations

If you want to remove a published graphics configuration or workstations from online but not from the configuration database, you can un-publish it. You can do so by selecting the Un-publish option from the context menu that appears when right-clicking on the item.

Setting displays as a Work in Progress

If changes made to a display are not ready to be published, the display can be set it as Work in Progress. When publishing displays that are set as a Work in Progress, they are automatically deselected in the Publish dialog. It is possible to still publish them by selecting the appropriate check box.
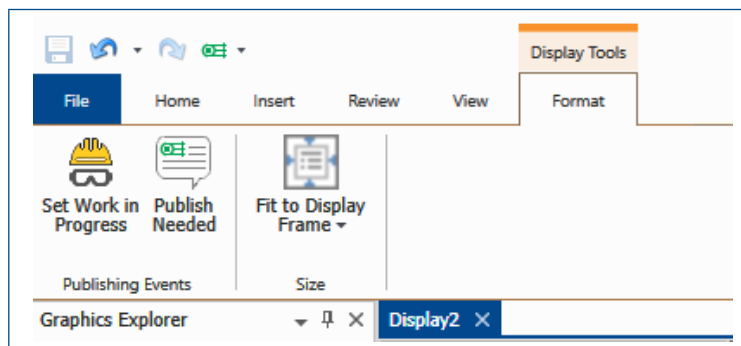

Figure 38. Set Work in Progress button in ribbon

Conversely, if a display needs to be published by someone else, the display can be set as Publish Needed, so others can do so when accessing the system, even from a different workstation.

Welcome to DeltaV Live

Working with Contextual Displays

A contextual display is a multi-use display that provides specific functionality such as a loop faceplate or a detail display. They are called in context for a module or parameter and allow you to create a purpose-built pop-up dialog for interactive use.

Each new contextual display is created with a base set of properties for consistency, which are used by out-of-the-box functions to pass contextual data to the display. The display can use this data to modify its behavior as desired, such as handling a Mode write versus and PID Setpoint write. Additional variables can be created to manage data once the display is open, along with interactions to allow the operator to manipulate control parameters.

Figure 39. Context properties for Contextual Display

The Graphics Configuration Pane displays these properties using their titles. Note that any scripting that references these parameters use their property name. The default properties are:

Name	Title	Description	Syntax	Data Type
Tag	Tag Description	Main contextual reference: Module name or specific parameter path	Dsp.tag	String
Context1 Context2 Context3 Context4	Context 1 Default Value	Four Context values are available for use. A default value can be defined if appropriate.	Dsp.Context1	String
N/A	Context 1 Description	A description field indicating use of Context 1, offline use only	N/A	String

Contextual displays open as a new window, and as such have a Window Title property which is animated. Typically, the Tag Description is used (dsp.tag), but it can be whatever text is desired. The size of the display will determine the available space to display the title.

Unlike GEM's, there are no additional custom properties in the contextual display, but you can define variables to serve as place holders for additional information. Variables can be animated or tied to standards and functions. For example, if you would like to add a company logo to your pop-up displays, create a variable linked to an Image Standard, and place this in the display.
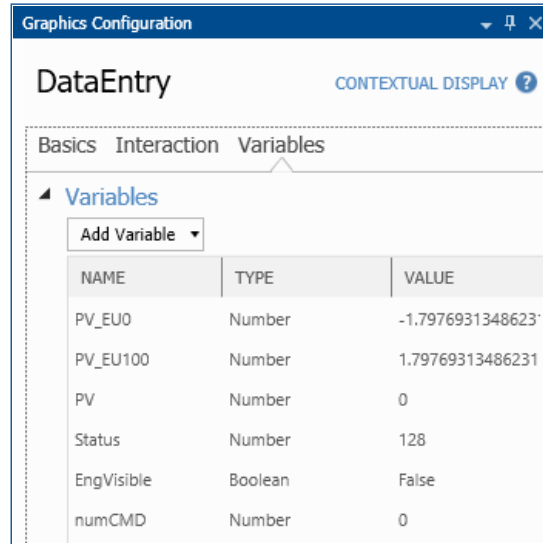

Figure 40. Adding Variables to Contextual Display

Finally, you can invoke custom scripts that trigger on display-based events, such as OnOpen or OnClose. These scripts would make use of the context properties and defined variables to achieve the functionality desired.
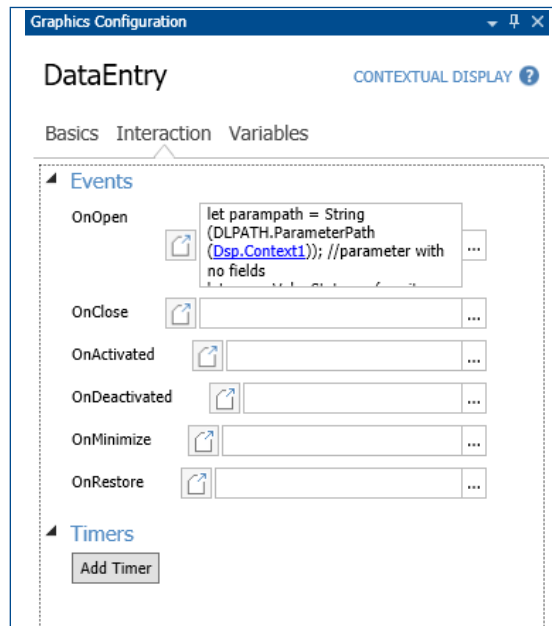

Figure 41. Adding Interactions to Contextual Display

The display can be resized and populated by data entry objects and GEMs to create a built for purpose dialog.

Next Steps

This document is only meant to introduce DeltaV Live. Books Online provides detailed information on how to do many additional things not mentioned in this whitepaper. Additional whitepapers will be produced in the future to expand on topics like working with GEMs, working with themes, Watch Areas, Live Snippets, embedded trends and managing workstations, among others.

Emerson
North America, Latin America:
☎ +1 800 833 8314 or
☎ +1 512 832 3774

Asia Pacific:
☎ +65 6777 8211

Europe, Middle East:
☎ +41 41 768 6111

🌐 **www.emerson.com/deltav**

The Emerson logo is a trademark and service mark of Emerson Electric Co. The DeltaV logo is a mark of one of the Emerson family of companies. All other marks are the property of their respective owners.

The contents of this publication are presented for informational purposes only, and while diligent efforts were made to ensure their accuracy, they are not to be construed as warranties or guarantees, express or implied, regarding the products or services described herein or their use or applicability. All sales are governed by our terms and conditions, which are available on request. We reserve the right to modify or improve the designs or specifications of our products at any time without notice.

EMERSON