



ACCELERATING
INNOVATION

Session 3-1089

To Live or Not to Live:

Transforming DeltaV Operate

into DeltaV Live

Disclaimer

The information and/or opinions expressed in this presentation are those of the authors and do not necessarily represent official policy or permission of Emerson or Emerson Exchange.

Important Reminders

Photography and audio/video recording is not permitted in any session, or in the exhibition areas, without press credentials or written permission from Emerson or Emerson Exchange.

Inquiries should be directed to:
EmersonExchange@Emerson.com



Benji Kidmose

Automation Lead Kirby North/South Canadian Natural
Northern Alberta Canada



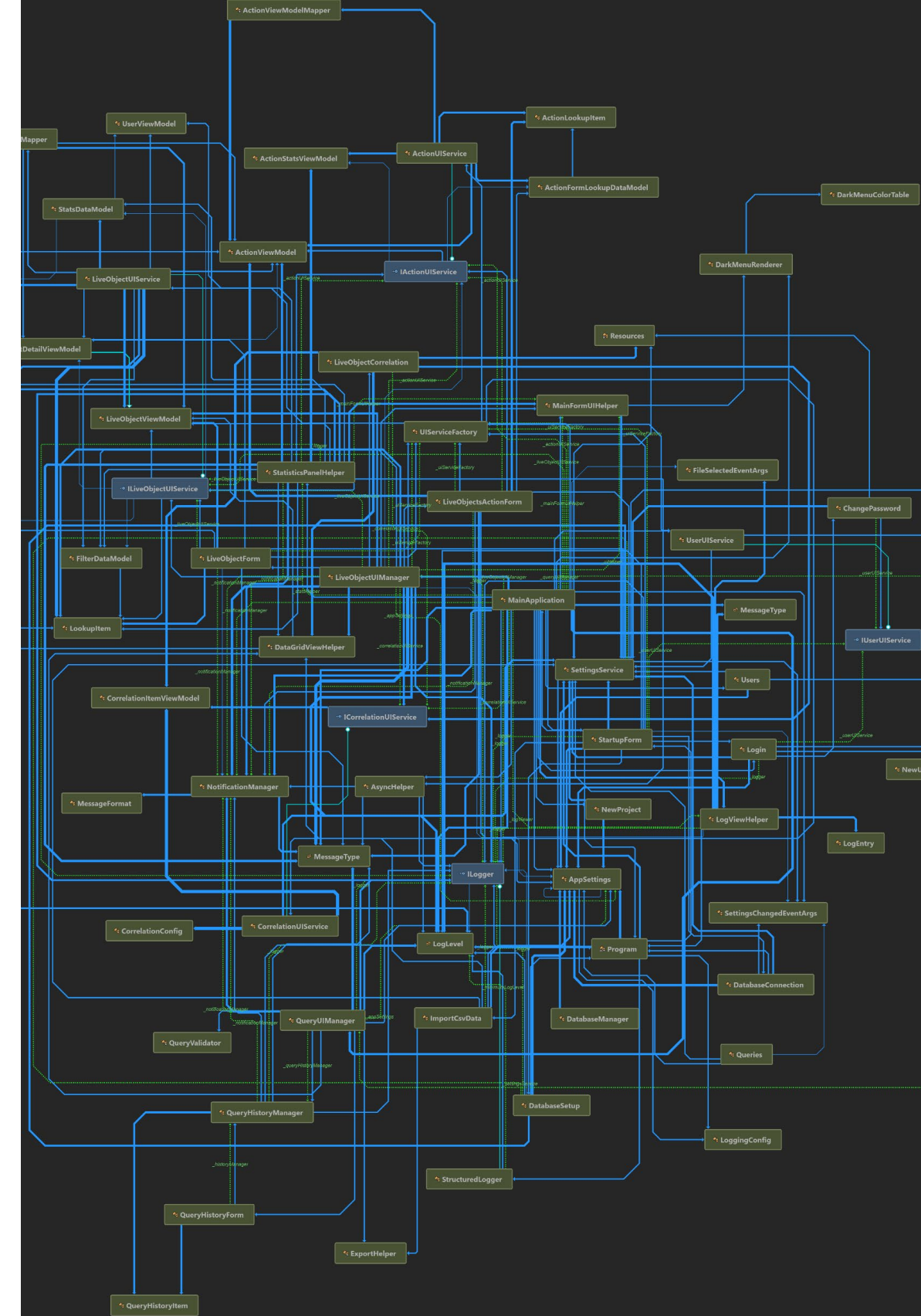
Andre Dicaire

Spartan Controls DeltaV Expert



Project Overview

- Project Scope
 - Operate to Live Migration
 - Two operating Steam Assisted Gravity Drainage (SAGD) facilities
 - Both running on V15.LTS
 - 26,000+ hard/soft combined I/O points
 - 7-person in-house automation team
 - 2 co-op Software Engineering students



Team Structure & Organization

- Team Composition
 - 7 full time Automation team members
 - 2 co-op Software Engineering Students on a year term
 - 1 member on 4/3 rotation
 - 8 members on 8/6 rotation
 - 1 dedicated Operations liaison
- Role Distribution
 - Senior members – Standards & Early Development
 - Junior members – Implementation
 - 1 Operations liaison making decision when required
 - 1 team member is final say and approval

Team Work



Business Challenge

Key Drivers

- No Immediate Business Case for Platform Change
 - Difficulty in quantifying immediate ROI
- End of Support for DeltaV Operate
 - Not being able to apply future upgrade past V16 and running in an un-supported state
- Cost Management via In-House Development
 - Leveraging internal resources to control expenses
- Balancing HCD with Operator Familiarity
 - Ensuring usability while introducing modern design practices

Strategic Benefits

- Enhanced Team Engagement and Ownership
 - Empowerment through direct involvement
- Improved Knowledge Retention
 - Minimize reliance on external consultants
- Tailored Custom Solution Development
 - Solutions precisely fit operational needs
- Long-Term Emerson Development
 - Continued improvements on product
- Early Operations Involvement
 - Reduced resistance, increased buy-in, smoother implementation

Project Constraints and Challenges

- Resource Management
 - Limited team size
 - Shift rotation complexities
 - Ongoing Operational duties
 - Ensuring stakeholder collaboration does not stall progress
- Technical Challenges
 - How do we manage a project of this size
 - Learning curve
 - Functionality differences
 - Changes to existing entities in Operate once they have been completed in Live (Change Management)
- Timeline Pressure
 - Self Imposed
 - 24-month parallel operation target



Lost Functionality In Live & Development Plan

- Identification Of Non-Reproducible Functionality
 - Potential for considerable time spent
 - Potential development in Control Studio
 - Potential Development outside of DeltaV software suites
 - May become a go-no-go situation if a solution cannot be found
 - May add a considerable expense to HMI migration

(General)

```
'Open the text file in a Read State
textFileRead = FreeFile
Open filePathRead For Input As textFileRead

'Store file content inside of a variable
fileContentRead = Input(LOF(textFileRead), textFileRead)

'Close the Text File
Close textFileRead

'Seperate out the lines of data
lineArray() = Split(fileContentRead, vbCrLf)

'Read data into an Array Variable
For x = LBound(lineArray) To UBound(lineArray)
    If Len(Trim(lineArray(x))) <> 0 Then
        tempArray = Split(lineArray(x), delimiter)
'Store data into an array
        col = UBound(tempArray)
'Readjust Array Bounds
        ReDim Preserve dataArray(col, rw)
'Store data in columns and rows
'Store line of data into Array variable
        For y = LBound(tempArray) To UBound(tempArray)
            dataArray(y, rw) = tempArray(y)
        Next y
        rw = rw + 1
    Next x

'Determine the size of the array on the columns of tags so that we know how
dataArrayRows = (UBound(dataArray, 2) - LBound(dataArray, 2) + 1)
'update the label to show how many items we will be reading
totalRows.Caption = dataArrayRows

textFileWrite = FreeFile

'Open the Writing TextFile
Open filePathWrite For Output As textFileWrite

For i = 0 To dataArrayRows - 1
    tag = dataArray(0, i)
    offlinePV = dataArray(1, i)
    PV = frsReadValue(tag)
    Write #textFileWrite, tag, PV, offlinePV
'Store data in columns and rows
'Print #TextFileWrite, PV; Tab(1)
    sleep (250)
    progress.Caption = i + 1
    On Error Resume Next
Next i

'Close the file for writing
Close textFileWrite
lblDone.Visible = True

Else
Exit Sub
```

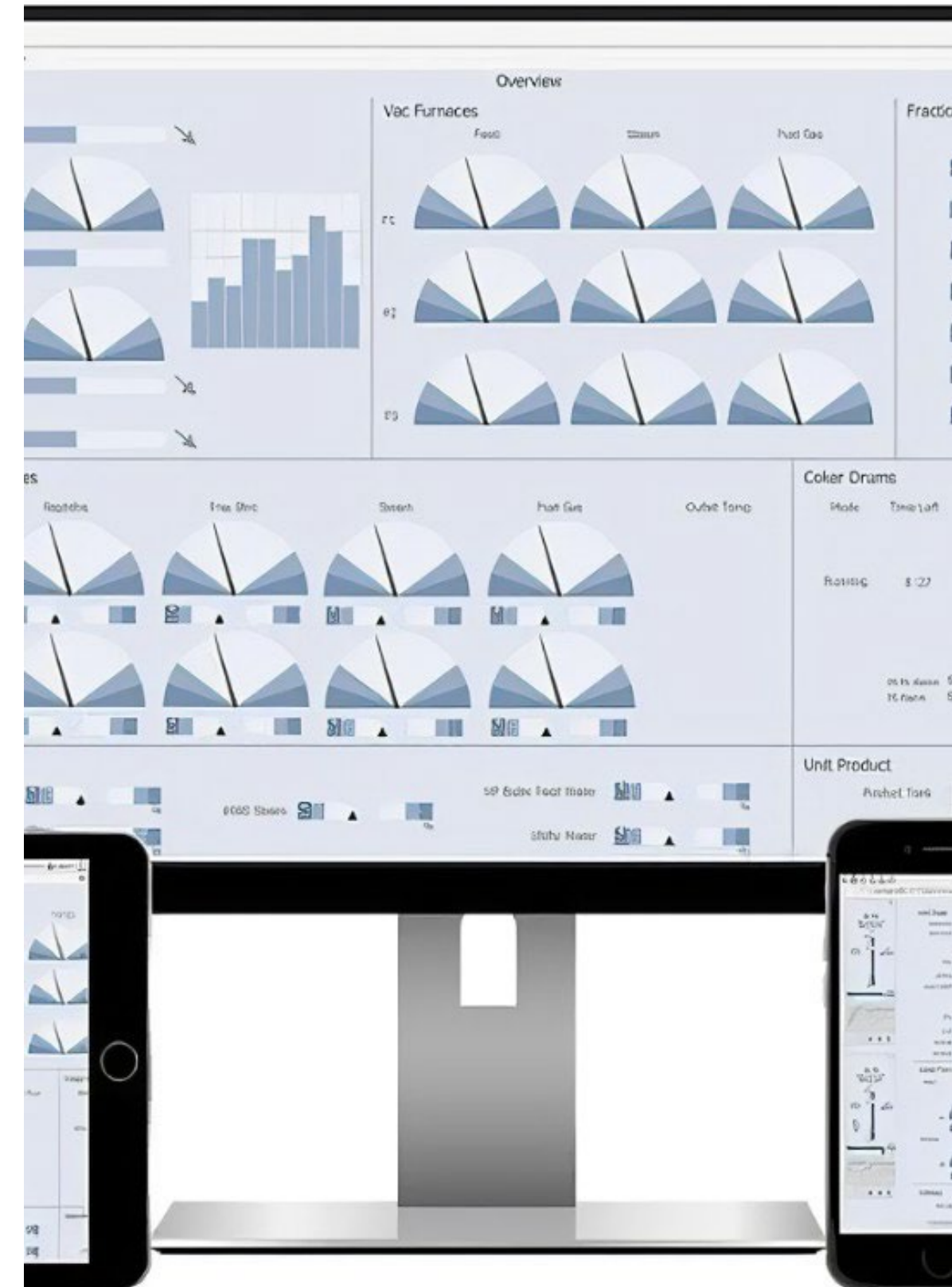
Resource Optimization

- Work Distribution
 - Dedicated project time
 - Protected development periods
 - Cross-shift handover process
- Task Management
 - Priority-based assignment
 - Skill-matched allocation
 - Progress tracking
- Project Time
 - Project time contributes to balancing innovation and familiarity.



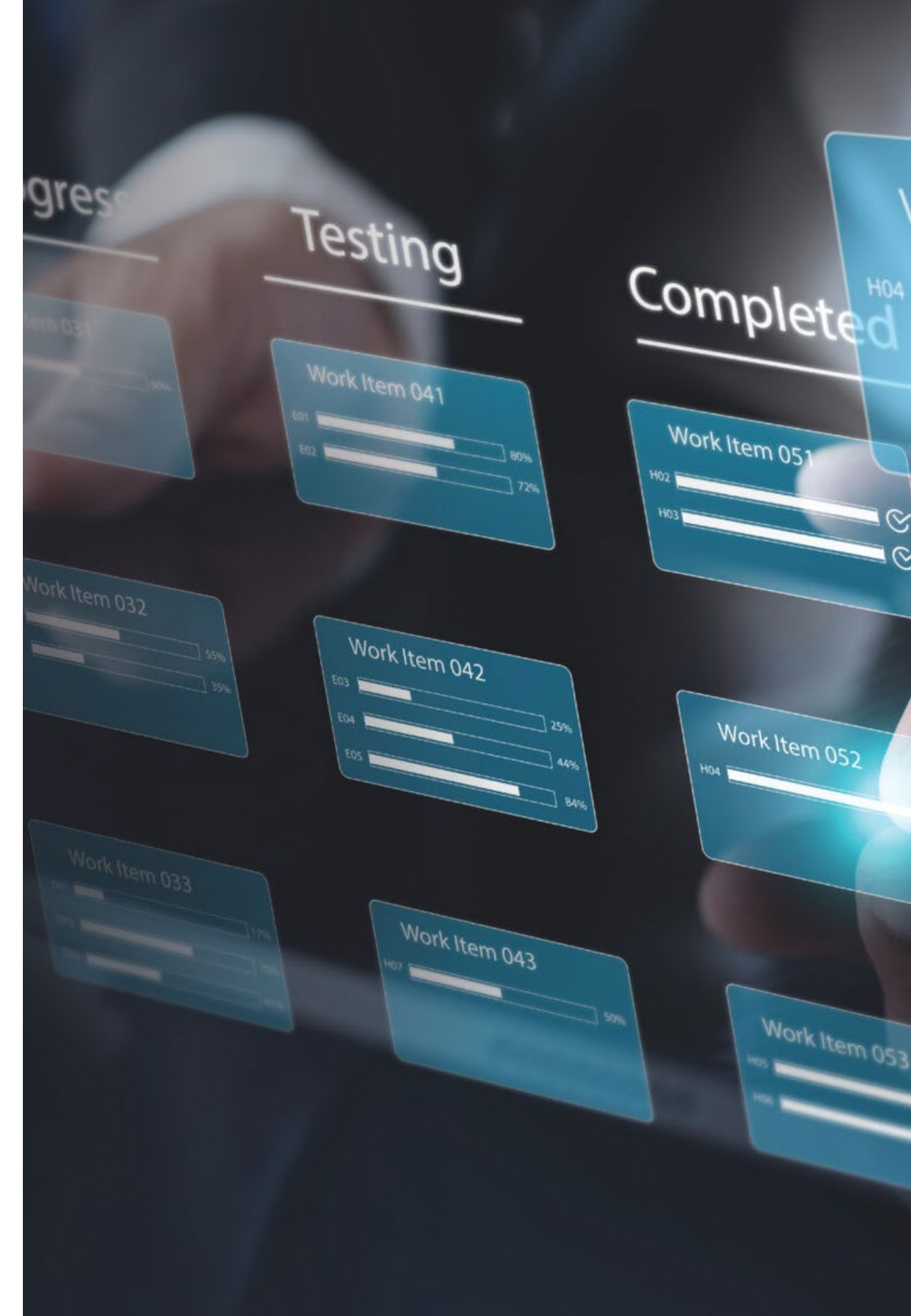
Development Methodology

- Order of Operations
 - Analyze Objects need to be duplicated from Operate into Live
 - Load all Operate Objects into the management software
 - Define as many Standards and Functions
 - GEM library creation
 - Faceplate/Detail standardization (with changes in classes)
 - Documentation requirements
 - Operator Involvement at the earliest stages
- Quality Control
 - Peer review process
 - Operations validation
 - Performance testing



Technical Implementation

- Development Focus
 - Define soft standards, colors, line hop types, font etc.
 - Identification and development of functionality developed in Operate that cannot be duplicated in Live
 - Static GEMs
 - L1, L2, L3, L4 Distribution
 - Static Graphics
 - Dynamic GEMs
 - Navigation structure
- Migration Strategy
 - Object-by-Object approach
 - Parallel testing
 - Operator feedback integration



Project Tracking Solution

- Custom Database Development
 - Initially a MS Access Database (Developed to start project in V14, but abandoned moving forward, waited till V15)
 - Migration to a SQL Server and C# solution using Entity Framework with .NET Framework 4.7
 - **Entity Framework (EF)** is a **.NET Object-Relational Mapper (ORM)** that simplifies database access by using C# objects instead of manual SQL queries.
 - Windows Form based front end for the UI
- Key Functions
 - Object tracking
 - Objects linked in relationship to existing Operate objects
 - Progress monitoring
 - Team assignments
 - Issues management
 - Allows Operator feedback to be captured

- + [table icon] dbo.ActionStatuses
- + [table icon] dbo.DeltavClasses
- + [table icon] dbo.DeltavModules
- + [table icon] dbo.DetailDTs
- + [table icon] dbo.DetailFPs
- + [table icon] dbo.DetailFunctions
- + [table icon] dbo.DetailGEMs
- + [table icon] dbo.DetailMDs
- + [table icon] dbo.DetailStandards
- + [table icon] dbo.FunctionBlockTypes
- + [table icon] dbo.Functions
- + [table icon] dbo.LiveDetailTypes
- + [table icon] dbo.LiveDisplayLevels
- + [table icon] dbo.LiveDisplaySetTypes
- + [table icon] dbo.LiveDisplayTypes
- + [table icon] dbo.LiveFacePlateTypes
- + [table icon] dbo.LiveFunctionTypes
- + [table icon] dbo.LiveGemTypes
- + [table icon] dbo.LiveLayoutTypes
- + [table icon] dbo.LiveObjectDeltavClasses
- + [table icon] dbo.LiveObjectDetailDisplays
- + [table icon] dbo.LiveObjectFaceplates
- + [table icon] dbo.LiveObjectFunctionBlocks
- + [table icon] dbo.LiveObjectFunctions
- + [table icon] dbo.LiveObjectGems
- + [table icon] dbo.LiveObjectOperates
- + [table icon] dbo.LiveObjects
- + [table icon] dbo.LiveObjectStandards
- + [table icon] dbo.LiveObjectTypes
- + [table icon] dbo.LiveStandardTypes
- + [table icon] dbo.LiveTemplateTypes
- + [table icon] dbo.LiveThemeTypes
- + [table icon] dbo.ObjectActions
- + [table icon] dbo.ObjectActionSequences
- + [table icon] dbo.ObjectActionTypes
- + [table icon] dbo.ObjectPriorities
- + [table icon] dbo.ObjectProjects

Live Migration Tracking Application

The screenshot displays the 'Live Migration Tracking Application' interface. It features a menu bar (File, View, Reports, Users, Projects, Help) and a sub-menu (Stats, Live Object Details, Queries, Event Log). The main area is divided into two sections: 'Object List' and 'Action Items'.

Object List Table:

ID	Created Date	Created By	Object Name	Object Type	Object Subtype	Status	Assigned To	Due Date	Project Name	Description
9	05/19/2025 14:34	admin	A101_L2_CGLNT	Main Display	Main Display-L2	New	admin	10/01/2025	Development	Main Display Area 101 Coagulent
10	05/19/2025 14:34	admin	A101_L2_DIL_REC	Main Display	Main Display-L2	New	admin	10/01/2025	Development	Main Display Area 101 Diluent Recovery
11	05/19/2025 14:34	admin	A101_L2_DILBIT_CLR	Main Display	Main Display-L2	Rejected	admin	10/01/2025	Development	Main Display Area 101 Dilbit Coolers
12	05/19/2025 14:34	admin	A101_L2_DILBIT_PMP	Main Display	Main Display-L2	New	admin	10/01/2025	Development	Main Display Area 101 Dilbit Pumps
13	05/19/2025 14:34	admin	A101_L2_DOILD_H2O	Main Display	Main Display-L2	New	admin	10/01/2025	Development	Main Display Area 101 De-Oiled Water
14	05/19/2025 14:34	admin	A101_L2_DSND	Main Display	Main Display-L2	New	admin	10/01/2025	Development	Main Display Area 101 Desanding
15	05/19/2025 14:34	admin	A101_L2_EMUL_BFW_EXCH	Main Display	Main Display-L2	New	admin	10/01/2025	Development	Main Display Area 101 Emulsion BFW Exchangers
16	05/19/2025 14:34	admin	A101_L2_FWKO	Main Display	Main Display-L2	New	admin	10/01/2025	Development	Main Display Area 101 FWKO
17	05/19/2025 14:34	admin	A101_L2_H2O	Main Display	Main Display-L2	New	admin	10/01/2025	Development	Main Display Area 101 Water
18	05/19/2025 14:34	admin	A101_L2_IGF	Main Display	Main Display-L2	New	admin	10/01/2025	Development	Main Display Area 101 Induced Gas Flotation
19	05/19/2025 14:34	admin	A101_L2_INLET	Main Display	Main Display-L2	New	admin	10/01/2025	Development	Main Display Area 101 Inlet Skid and ESD
20	05/19/2025 14:34	admin	A101_L2_OIL_TRAIN	Main Display	Main Display-L2	New	admin	10/01/2025	Development	Main Display Area 101 Oile Train
21	05/19/2025 15:43	admin	KBP_A_AI_fp	Faceplate	Analog In Alarms O...	In Progress	benjik	11/12/2025	Development	AI Faceplate with Alarms Only

Action Items Table:

Action ID	Object ID	Title	Description	Type	Priority	Status	Current Owner	Complete
6	12	Create Static Display	Create the Display and draw all static images	Approved	Medium	New	admin	<input type="checkbox"/>
7	12	Graphic Created	Graphic Created with all Static Objects	Create	Medium	In Progress	benjik	<input type="checkbox"/>
8	12	Operations Approval of Static	Operations to Approve Static	Request For Approval	Critical	On Hold	admin	<input type="checkbox"/>

The right-hand side of the interface shows a detailed view for the selected object (ID 12):

- Live Object ID: 12
- Live Object Name: A101_L2_DILBIT_PMP
- Live Object Description: Main Display Area 101 Dilbit Pun
- Live Object Type: Main Display
- Live Object Sub-Type: Main Display-L2
- Operate Correlation: A101_L2_DILBIT_PMP
- Assigned To: [Empty field]
- Detailed Description: [Empty text area]

At the bottom, there is a status bar: "Monday, May 19, 2025 03:57:07 PM atabase Path Connected to: LiveMigration Logged On User Logged in as: admin (ID: 1)"

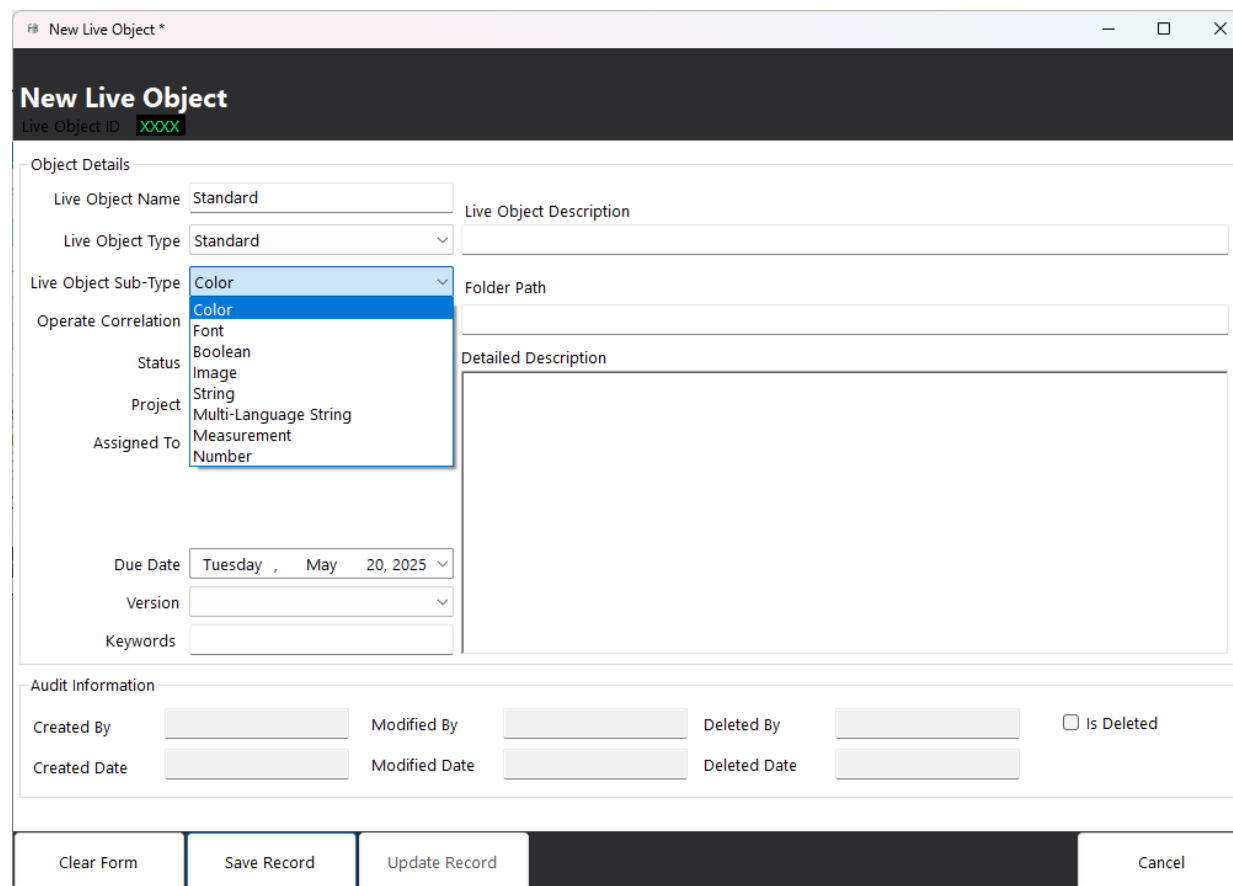
Centralized Management Interface:

- Provides a clear, structured view of each **Live Object** and its associated **Actions**, allowing easy navigation and detailed management from a single point.

Detailed Action Information:

- Displays comprehensive details including action title, type, priority, status, current owner, completion status, and due dates

Live Migration Tracking Application



New Live Object
Live Object ID: XXXX

Object Details

Live Object Name: Standard
Live Object Description:

Live Object Type: Standard
Live Object Sub-Type: Color
Folder Path:

Operate Correlation: **Color**
Font
Boolean
Image
String
Project
Multi-Language String
Assigned To: Measurement
Number

Status:
Detailed Description:

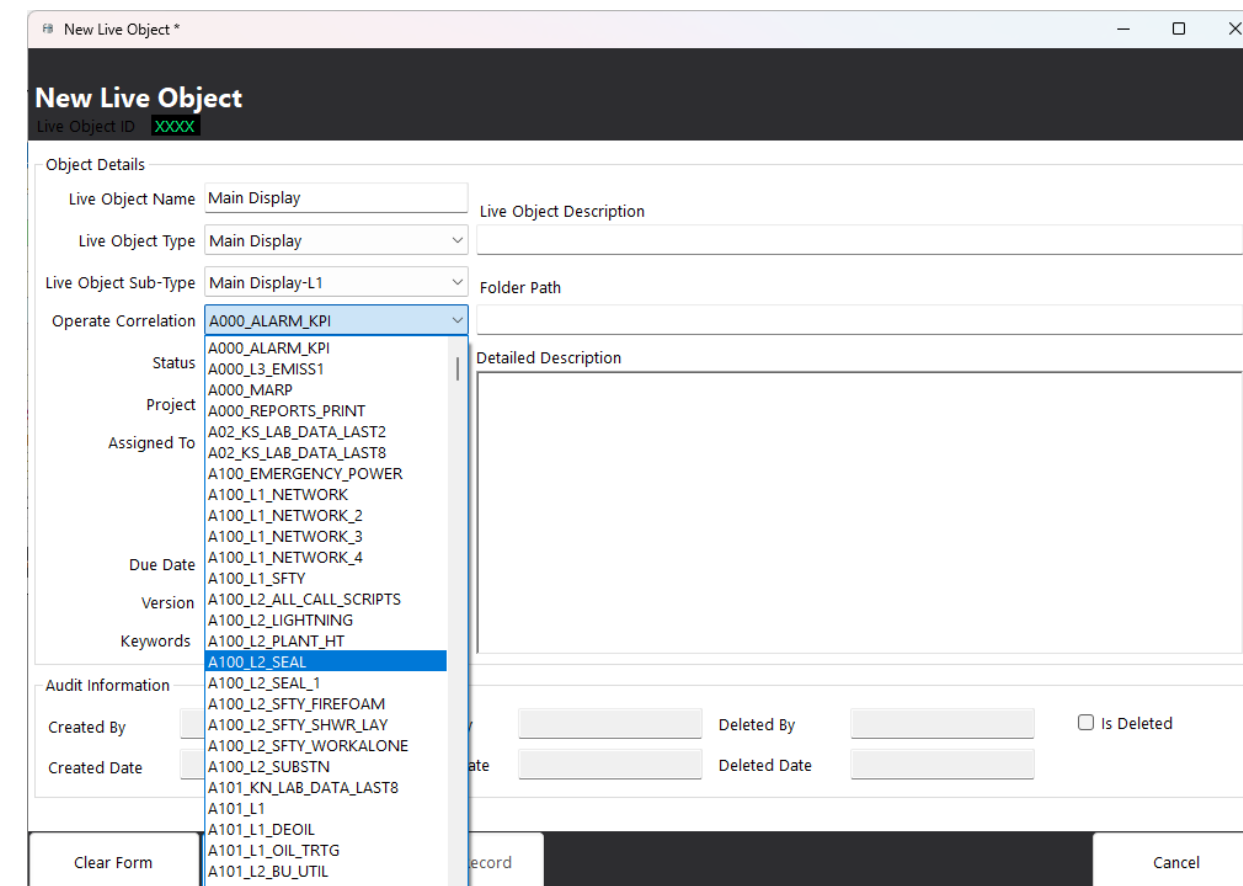
Due Date: Tuesday, May 20, 2025
Version:
Keywords:

Audit Information

Created By: Modified By: Deleted By: Is Deleted

Created Date: Modified Date: Deleted Date:

Clear Form Save Record Update Record Cancel



New Live Object
Live Object ID: XXXX

Object Details

Live Object Name: Main Display
Live Object Description:

Live Object Type: Main Display
Live Object Sub-Type: Main Display-L1
Folder Path:

Operate Correlation: A000_ALARM_KPI
Status: A000_ALARM_KPI
Project: A000_MARP
Assigned To: A02_KS_LAB_DATA_LAST2
A02_KS_LAB_DATA_LAST8
A100_EMERGENCY_POWER
A100_L1_NETWORK
A100_L1_NETWORK_2
A100_L1_NETWORK_3
A100_L1_NETWORK_4
A100_L1_SFTY
Due Date: A100_L2_ALL_CALL_SCRIPTS
Version: A100_L2_LIGHTNING
Keywords: A100_L2_PLANT_HT
A100_L2_SEAL
A100_L2_SEAL_1
A100_L2_SFTY_FIREFOAM
A100_L2_SFTY_SHWR_LAY
A100_L2_SFTY_WORKALONE
A100_L2_SUBSTN
A101_KN_LAB_DATA_LAST8
A101_L1
A101_L1_DEOIL
A101_L1_OIL_TRTG
A101_L2_BU_UTIL
A101_L2_CFG_INT

Status:
Detailed Description:

Due Date:
Version:
Keywords:

Audit Information

Created By: Modified By: Deleted By: Is Deleted

Created Date: Modified Date: Deleted Date:

Clear Form Save Record Update Record Cancel

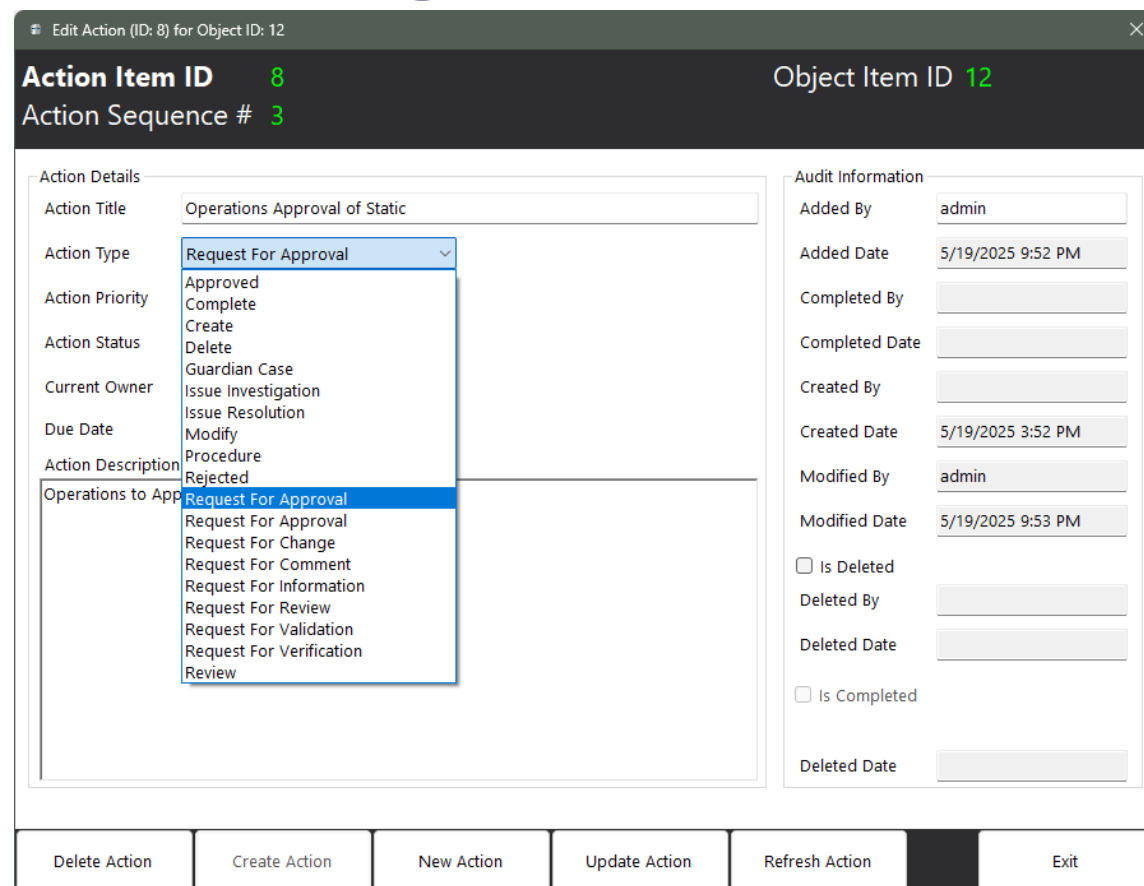
New Live Object Creation:

- Can choose an Object type and then determine sub-type of the Object

Link Object in a 1 to 1 correlation:

- All Object types from Operate are loaded into database so that each time an Object type is chose and 1 to 1 relationship can be defined for the Live Object to Operate Object

Live Migration Tracking Application



Edit Action (ID: 8) for Object ID: 12

Action Item ID 8 **Object Item ID 12**
Action Sequence # 3

Action Details

Action Title: Operations Approval of Static

Action Type: Request For Approval (dropdown menu open)

Action Priority: Approved

Action Status: Create

Current Owner: Delete

Due Date: Guardian Case

Action Description: Issue Investigation

Operations to Approve: Issue Resolution

Audit Information

Added By: admin

Added Date: 5/19/2025 9:52 PM

Completed By: [empty]

Completed Date: [empty]

Created By: [empty]

Created Date: 5/19/2025 3:52 PM

Modified By: admin

Modified Date: 5/19/2025 9:53 PM

Is Deleted

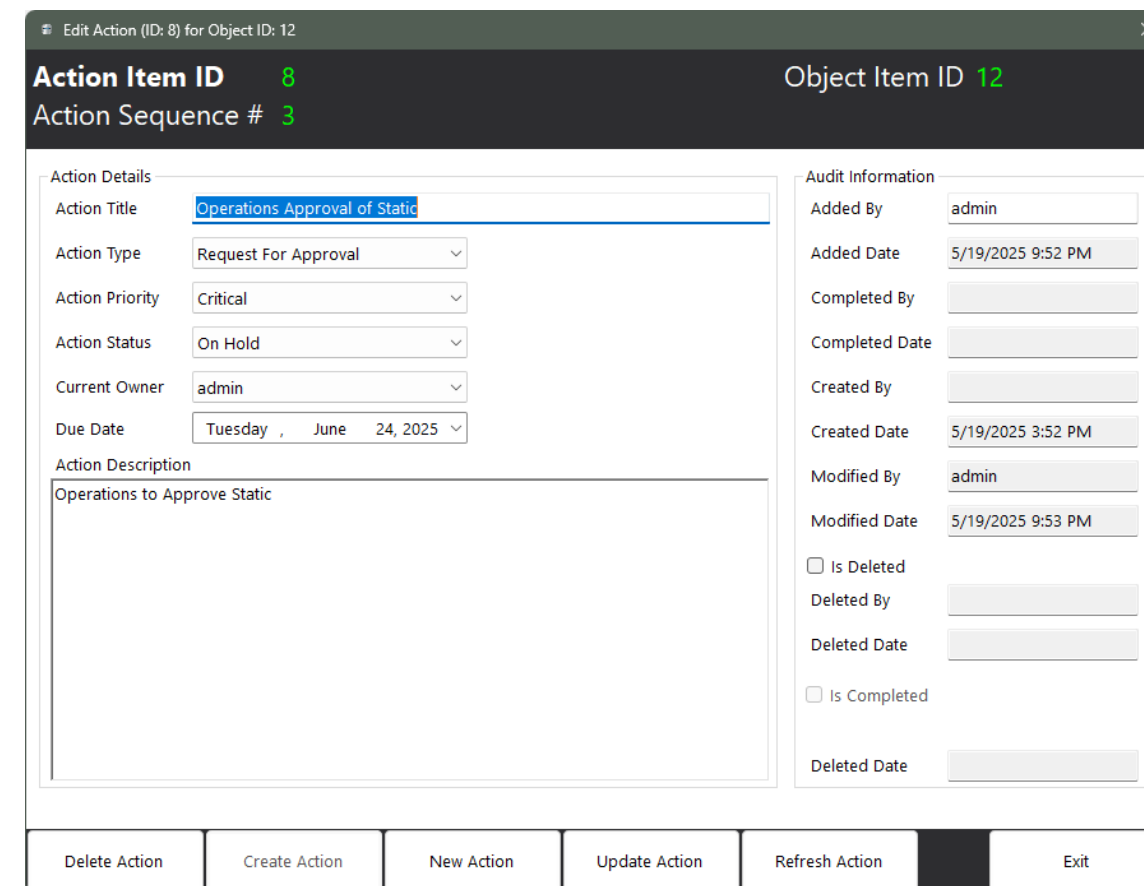
Deleted By: [empty]

Deleted Date: [empty]

Is Completed

Deleted Date: [empty]

Buttons: Delete Action, Create Action, New Action, Update Action, Refresh Action, Exit



Edit Action (ID: 8) for Object ID: 12

Action Item ID 8 **Object Item ID 12**
Action Sequence # 3

Action Details

Action Title: Operations Approval of Static

Action Type: Request For Approval

Action Priority: Critical

Action Status: On Hold

Current Owner: admin

Due Date: Tuesday, June 24, 2025

Action Description: Operations to Approve Static

Audit Information

Added By: admin

Added Date: 5/19/2025 9:52 PM

Completed By: [empty]

Completed Date: [empty]

Created By: [empty]

Created Date: 5/19/2025 3:52 PM

Modified By: admin

Modified Date: 5/19/2025 9:53 PM

Is Deleted

Deleted By: [empty]

Deleted Date: [empty]

Is Completed

Deleted Date: [empty]

Buttons: Delete Action, Create Action, New Action, Update Action, Refresh Action, Exit

Sequential Action Numbering:

- Automatically generates and maintains sequential numbering for actions tied to each object, ensuring systematic organization and easy tracking and referencing of Action ID's
- Provides clarity in task prioritization and historical tracking.

Audit and Accountability:

- Comprehensive auditing captures detailed history including who created, modified, completed, or deleted each action, along with timestamps for each
- Facilitates transparent communication and simplifies auditing and compliance reviews.

Live Migration Tracking Application

Live Migration Project - Object Correlation

Correlations

GEMS Standards Functions Modules Class Modules Function Blocks Operate Displays Operate Faceplates Operate Detail Displays Dynamos Charts User Globals

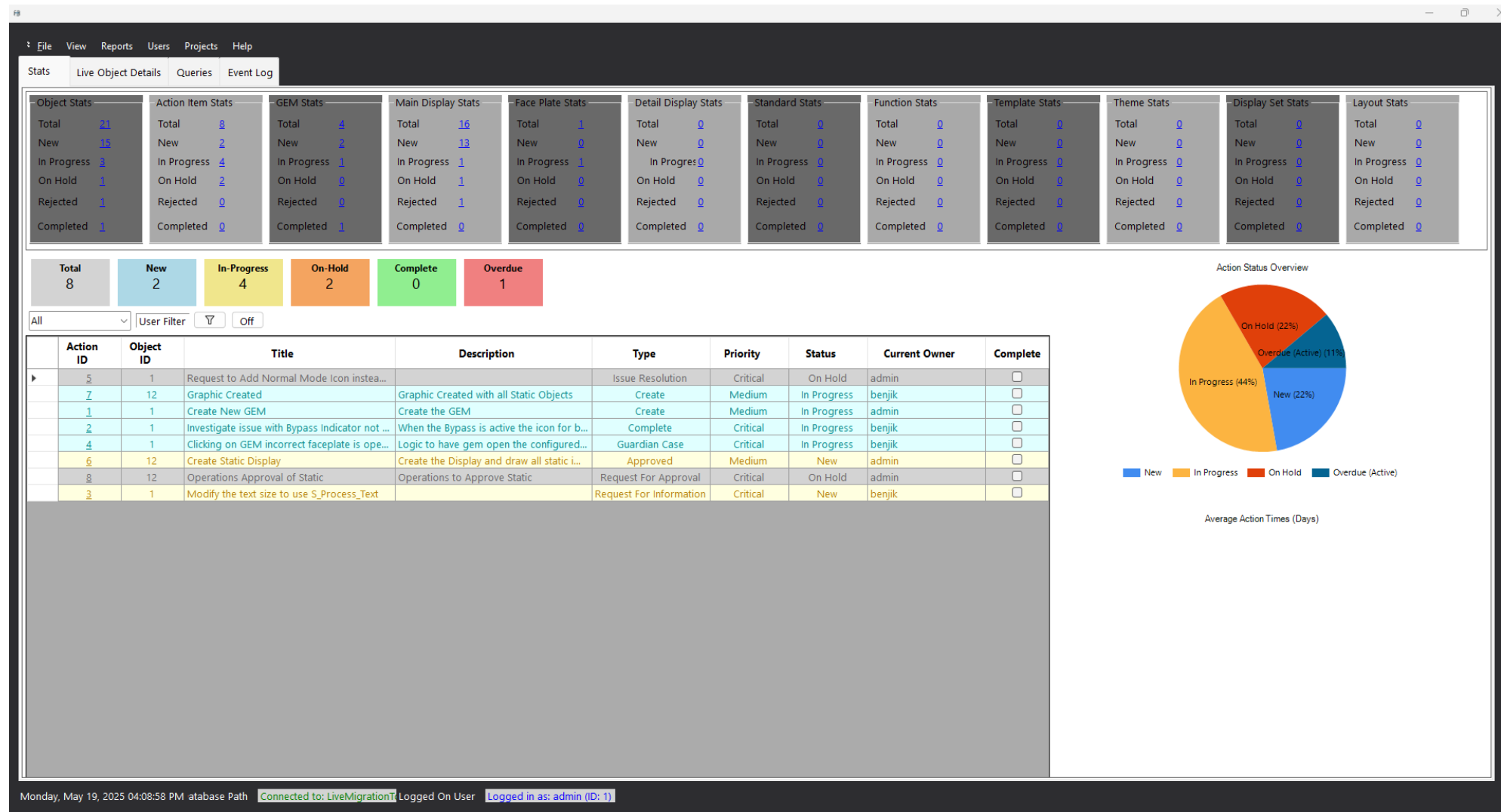
Show Only Assigned

Id	Assigned	Name
2	<input type="checkbox"/>	ABS
3	<input type="checkbox"/>	ACT
4	<input type="checkbox"/>	ADD
5	<input type="checkbox"/>	AGA_SI
6	<input type="checkbox"/>	AGA_US
7	<input checked="" type="checkbox"/>	AI
8	<input checked="" type="checkbox"/>	ALARM
9	<input type="checkbox"/>	AND
10	<input type="checkbox"/>	AO
11	<input type="checkbox"/>	ARTHM
12	<input type="checkbox"/>	AT
13	<input type="checkbox"/>	AVTR
14	<input type="checkbox"/>	BDE
15	<input type="checkbox"/>	BFI
16	<input type="checkbox"/>	BFO
17	<input type="checkbox"/>	BG
18	<input type="checkbox"/>	CALC
19	<input type="checkbox"/>	CEM
20	<input type="checkbox"/>	CMP
21	<input type="checkbox"/>	CND
22	<input type="checkbox"/>	CTI SI

Relationships Multiple-to-Multiple Associations:

- Each **Live Object** (such as GEMS, Standards, Functions, Modules, etc.) can be associated with multiple correlated entities. (Sub-Gems to Gems)
- Each correlated entity (like Function Blocks, can be associated with multiple **Live Objects**.
- Completed through the use of Junction Tables

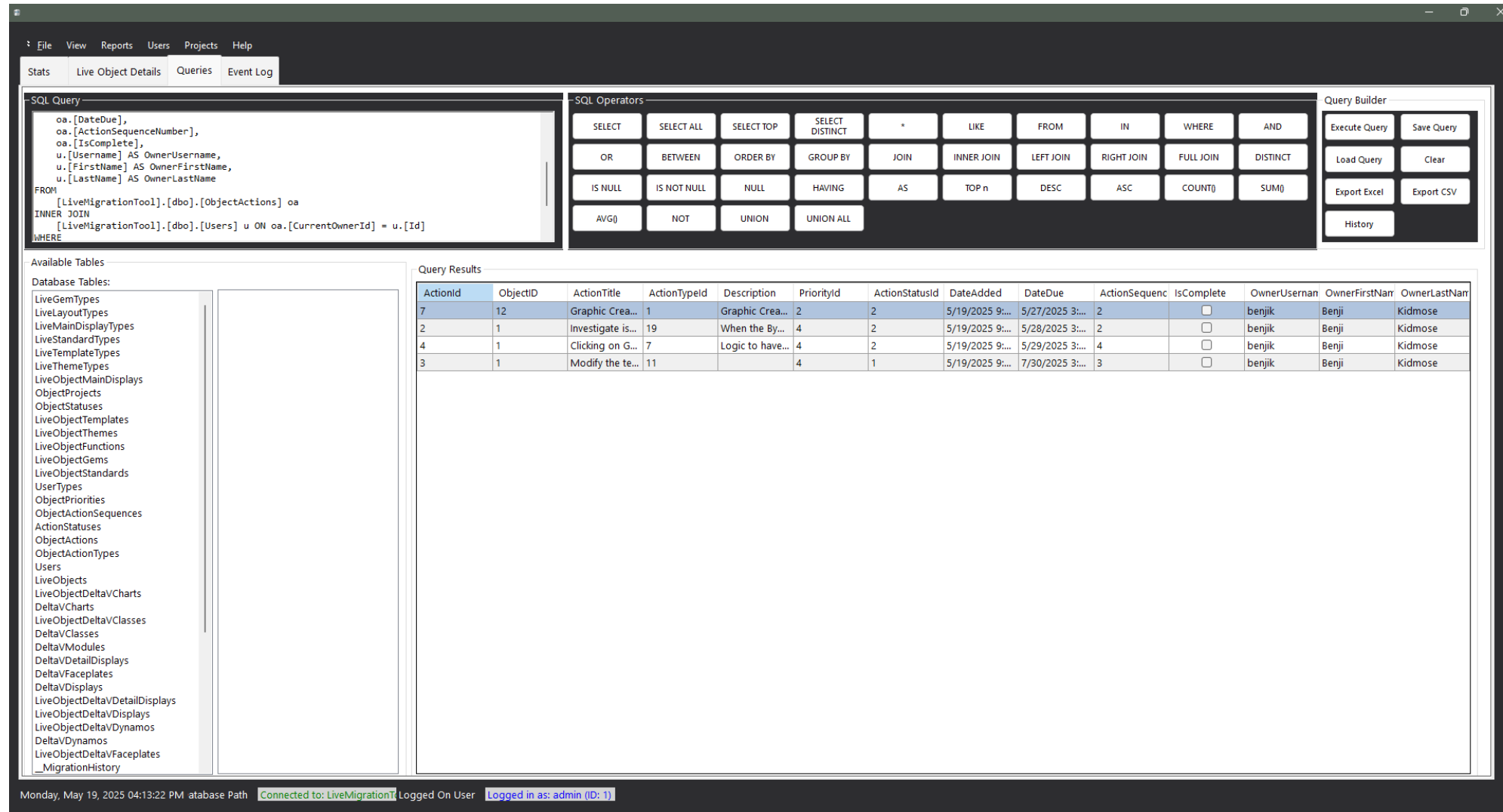
Live Migration Tracking Application



Real-Time Status and Progress Tracking:

- Immediate visibility into the current state and progress of each action (e.g., New, In Progress, Completed, On Hold, Overdue)
- Visual indicators and data fields allow users to quickly assess the urgency and status of tasks, supporting timely decision-making.
- Tiles to indicate the number of Actions and the current Status

Live Migration Tracking Application



The screenshot displays the application's interface with the following components:

- SQL Query Editor:** Contains the following query:


```

      SELECT
      oa.[DateDue],
      oa.[ActionSequenceNumber],
      oa.[IsComplete],
      u.[Username] AS OwnerUsername,
      u.[FirstName] AS OwnerFirstName,
      u.[LastName] AS OwnerLastName
      FROM
      [LiveMigrationTool1].[dbo].[ObjectActions] oa
      INNER JOIN
      [LiveMigrationTool1].[dbo].[Users] u ON oa.[CurrentOwnerId] = u.[Id]
      WHERE
      
```
- SQL Operators:** A grid of buttons for SQL keywords and operators including SELECT, FROM, WHERE, AND, JOIN, ORDER BY, GROUP BY, HAVING, AS, TOP n, DESC, ASC, COUNT(), SUM(), AVG(), NOT, UNION, and UNION ALL.
- Query Builder:** A panel with buttons for 'Execute Query', 'Save Query', 'Load Query', 'Clear', 'Export Excel', 'Export CSV', and 'History'.
- Available Tables:** A list of database tables on the left side of the interface.
- Query Results:** A table displaying the results of the query:

ActionId	ObjectID	ActionTitle	ActionTypeId	Description	PriorityId	ActionStatusId	DateAdded	DateDue	ActionSequenc	IsComplete	OwnerUsernan	OwnerFirstNan	OwnerLastNan
7	12	Graphic Crea...	1	Graphic Crea...	2	2	5/19/2025 9:...	5/27/2025 3:...	2	<input type="checkbox"/>	benjik	Benji	Kidmose
2	1	Investigate is...	19	When the By...	4	2	5/19/2025 9:...	5/28/2025 3:...	2	<input type="checkbox"/>	benjik	Benji	Kidmose
4	1	Clicking on G...	7	Logic to have...	4	2	5/19/2025 9:...	5/29/2025 3:...	4	<input type="checkbox"/>	benjik	Benji	Kidmose
3	1	Modify the te...	11		4	1	5/19/2025 9:...	7/30/2025 3:...	3	<input type="checkbox"/>	benjik	Benji	Kidmose

Monday, May 19, 2025 04:13:22 PM atabase Path Connected to: LiveMigration7 Logged On User Logged in as: admin (ID: 1)

- **Interactive Database Query Interface:**

- Provides an intuitive environment for building and executing SQL queries directly within the application.
- Empowers users to quickly retrieve, manipulate, and analyze real-time data without needing external database tools.

- **Advanced Query Building:**

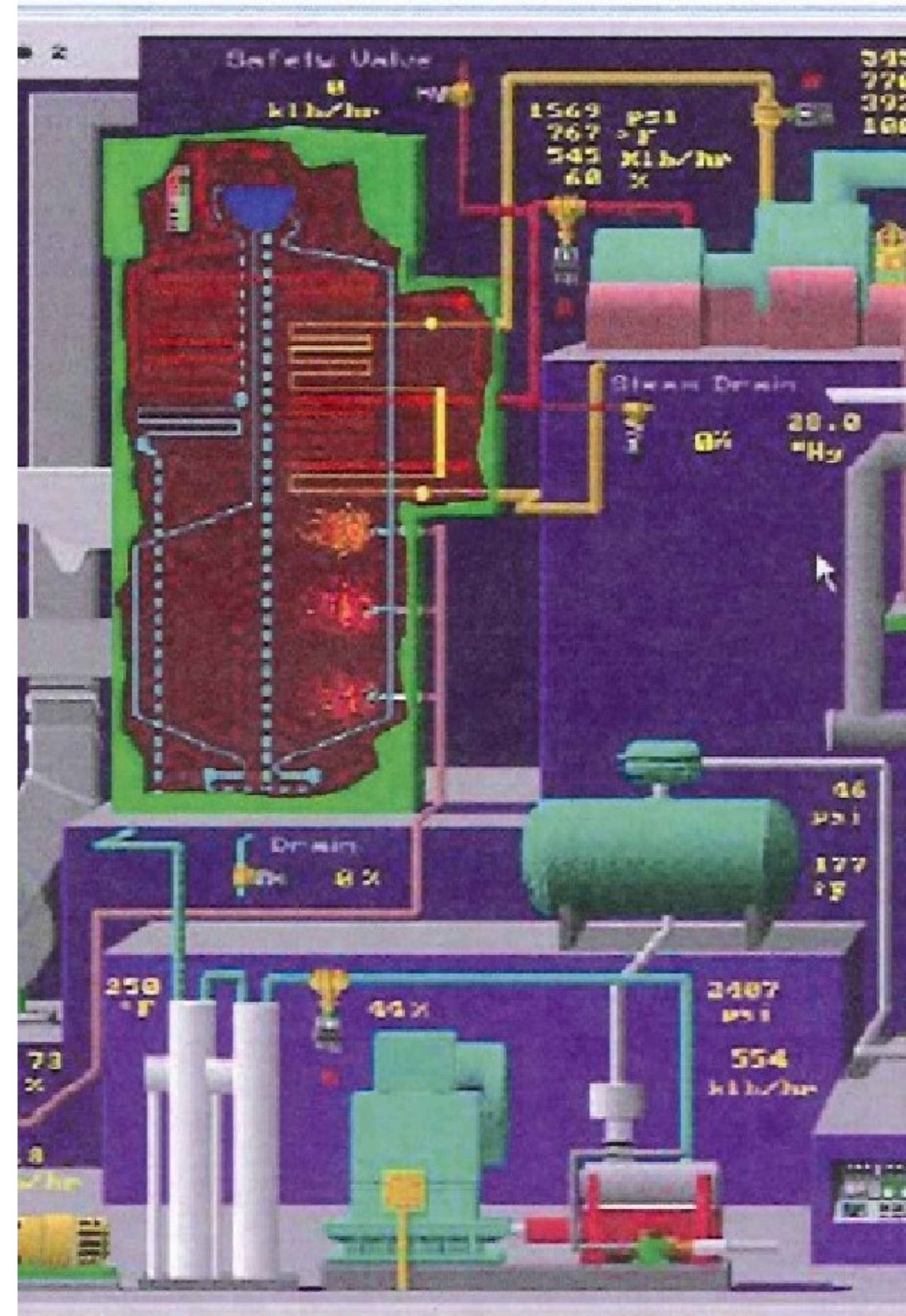
- Supports complex query creation through intuitive button-driven SQL operations like SELECT, INNER JOIN, LEFT JOIN, WHERE conditions, ORDER BY, GROUP BY, and aggregation functions (e.g., AVG, COUNT, SUM).
- Enables both novice and experienced users to rapidly build accurate queries.

Migration vs Modernization

Change is inevitable...

The move to Live is an opportunity for improvement.

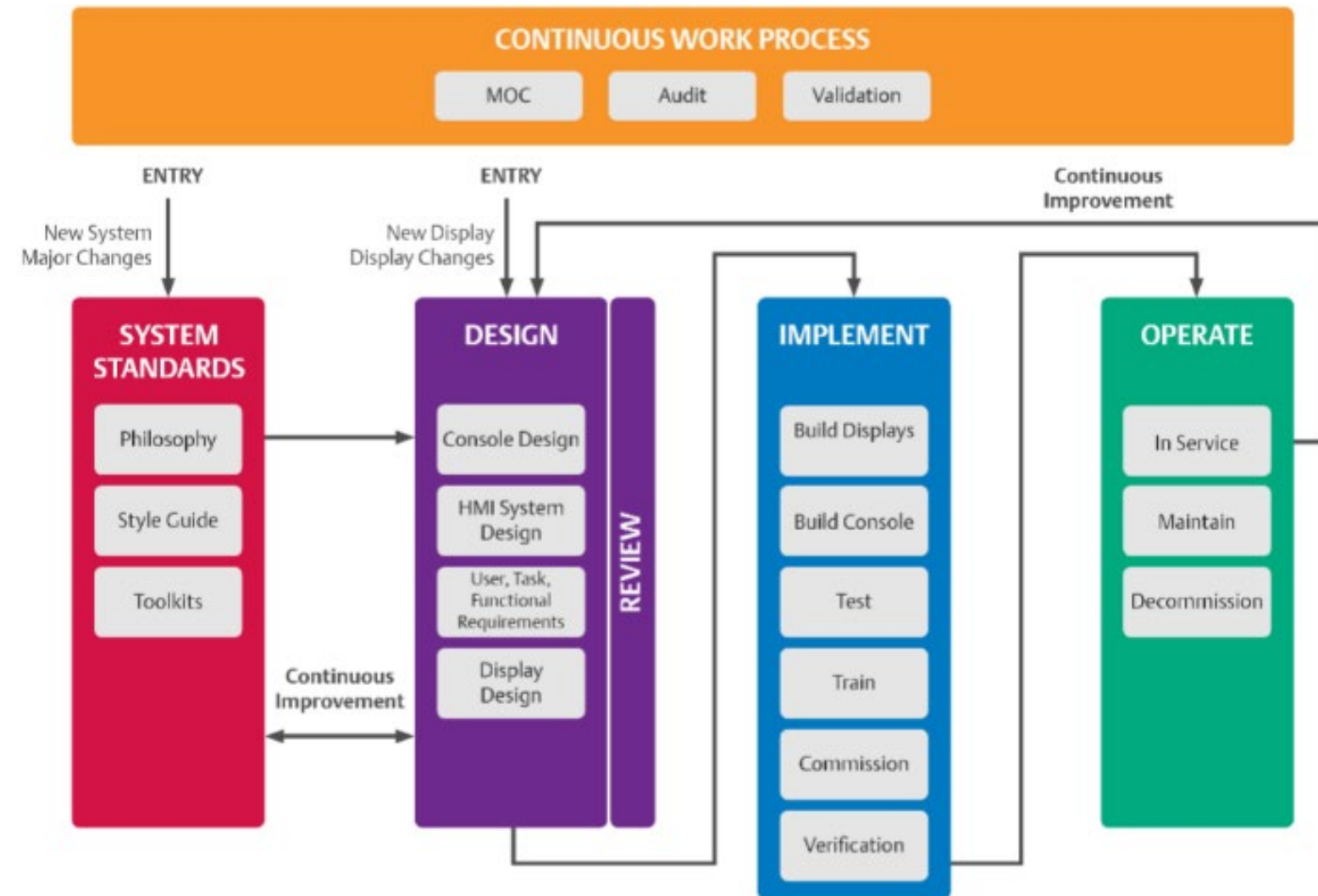
- Class based Objects for ease of maintenance and Change management
- Vector based Displays based on HTML5 bring forward looking options with resizable layouts and capabilities that support a mobile environment on the Edge
- Native Navigation hierarchy improves long term maintainability(no custom code)
- Typescript language for customization replaces VBA, and brings new Scripting options like Library Functions



Review and Enhance

Phased approach

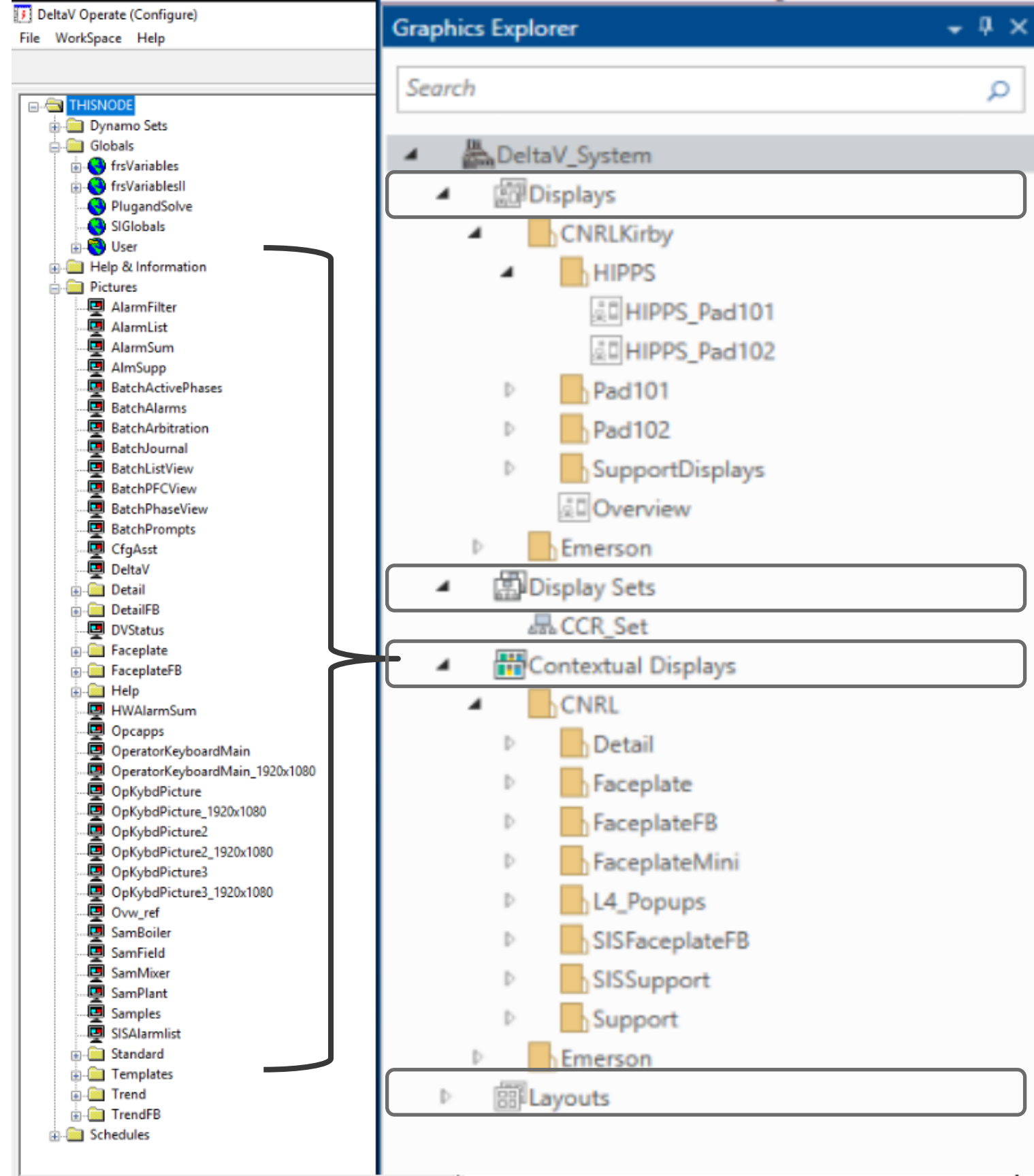
- Phase 1: Perform an Audit to define scope
 - Collect Usage Metrics
 - Review Display Usage and workflows
 - Evaluate custom solutions to identify/develop
 - Present Findings Display hierarchy and optimizations
 - Develop HMI Toolkit/Library for use in execution phase
- Phase 2: Execute Modernization plan
 - Build new displays, Migrate existing
 - Develop/Test custom solutions
 - Validate new environment with Operations
 - Transition to new HMI



Graphics Studio review

Display management in a central Database

- Display Folder tree structure to manage objects
- Display Set Navigation matching ISA 101 Hierarchy
- Contextual Displays and other L4 type graphics
- Console layout tool for managing monitor space

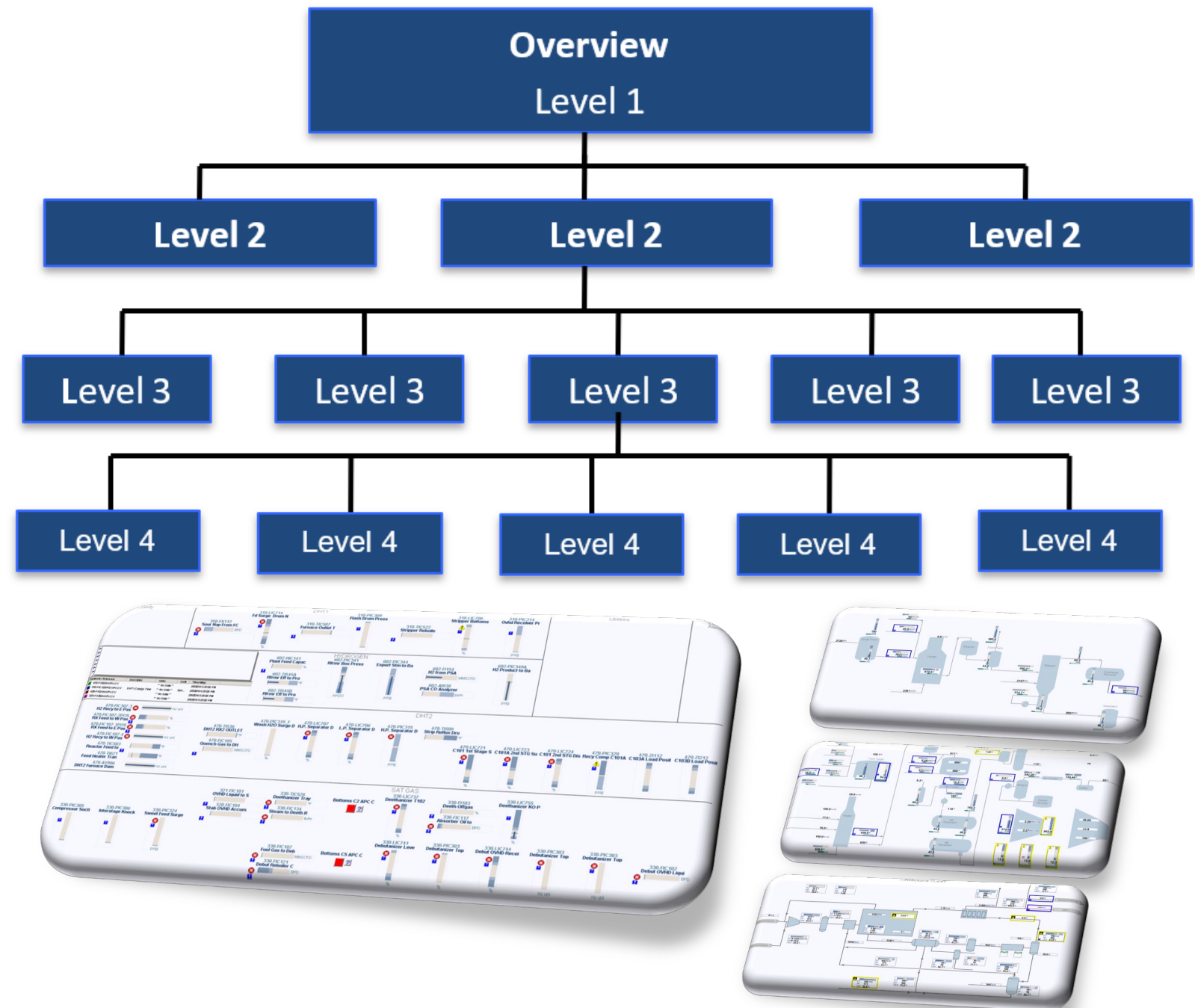


ISA 101.01 HMI Standard

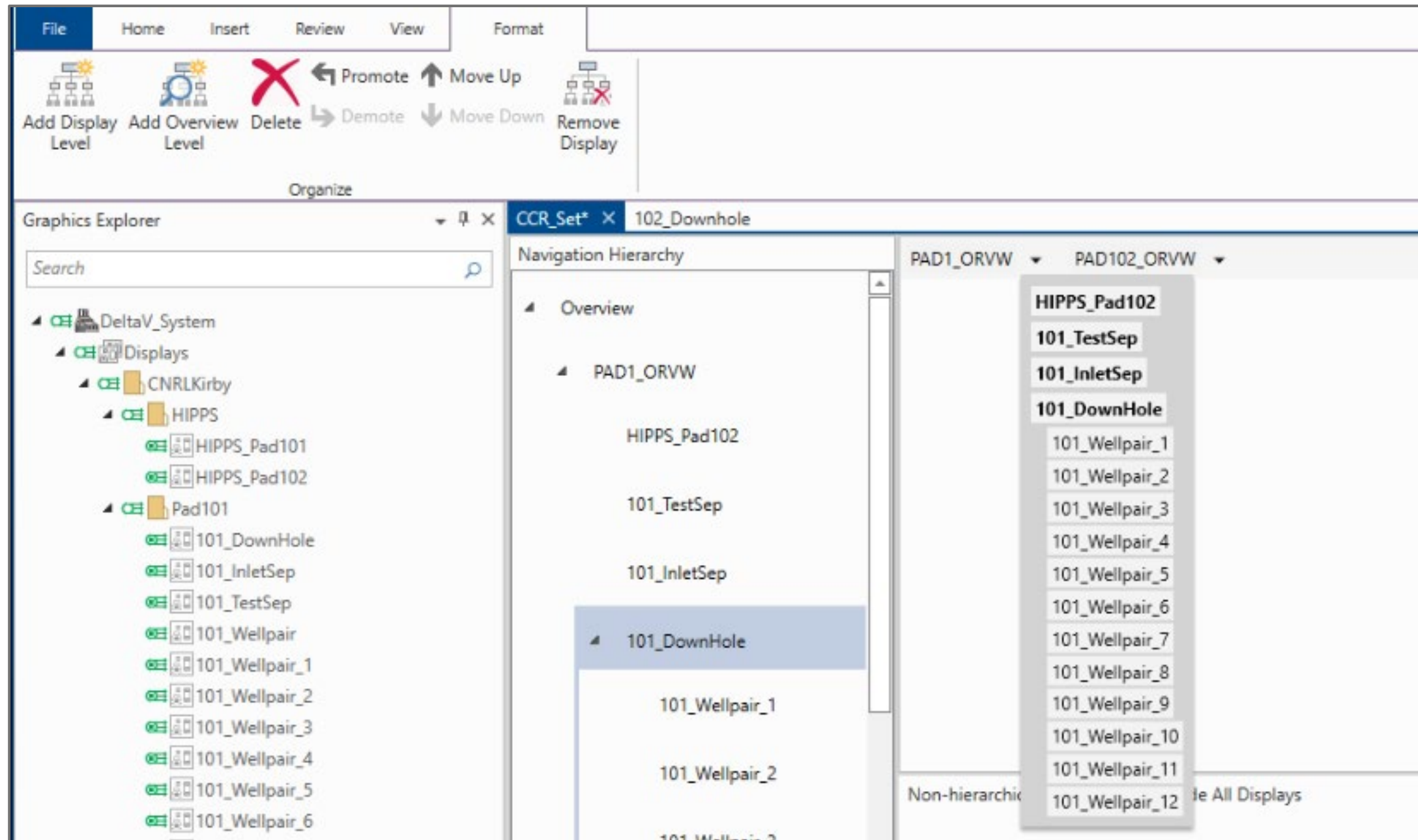
AMERICAN NATIONAL STANDARD

ANSI/ISA-101.01-2015

Human Machine Interfaces For
Process Automation Systems



Drag and Drop Navigation Hierarchy



The screenshot displays a software interface for managing a navigation hierarchy. At the top, a ribbon contains the 'Organize' group with the following options: Add Display Level, Add Overview Level, Delete, Promote, Demote, Move Up, Move Down, and Remove Display. Below the ribbon, the 'Graphics Explorer' pane shows a tree structure of the system components:

- DeltaV_System
 - Displays
 - CNRLKirby
 - HIPPS
 - HIPPS_Pad101
 - HIPPS_Pad102
 - Pad101
 - 101_DownHole
 - 101_InletSep
 - 101_TestSep
 - 101_Wellpair
 - 101_Wellpair_1
 - 101_Wellpair_2
 - 101_Wellpair_3
 - 101_Wellpair_4
 - 101_Wellpair_5
 - 101_Wellpair_6

The 'Navigation Hierarchy' pane shows the current view structure:

- Overview
 - PAD1_ORVW
 - HIPPS_Pad102
 - 101_TestSep
 - 101_InletSep
 - 101_DownHole (selected)

The detailed view of the selected '101_DownHole' display shows a list of wellpairs:

- HIPPS_Pad102
- 101_TestSep
- 101_InletSep
- 101_DownHole
 - 101_Wellpair_1
 - 101_Wellpair_2
 - 101_Wellpair_3
 - 101_Wellpair_4
 - 101_Wellpair_5
 - 101_Wellpair_6
 - 101_Wellpair_7
 - 101_Wellpair_8
 - 101_Wellpair_9
 - 101_Wellpair_10
 - 101_Wellpair_11
 - 101_Wellpair_12

At the bottom of the detailed view, there are buttons for 'Non-hierarchical' and 'Show All Displays'.

Display Frame Navigation

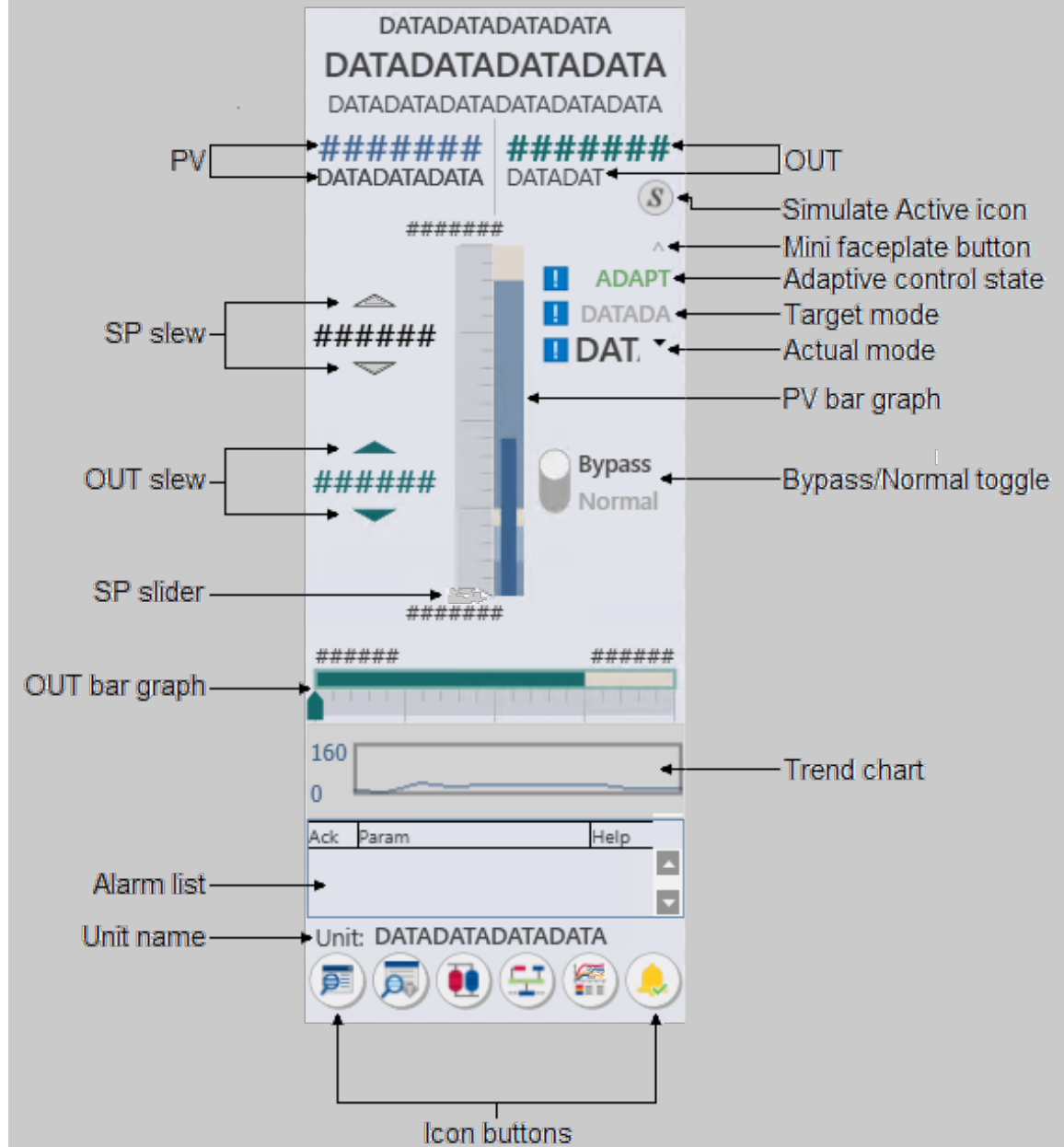
The screenshot displays a software interface with a navigation bar at the top. The navigation bar contains several icons: a magnifying glass (Search), a heart with a pulse line (Errors), a wrench (Tools), a circular arrow (Refresh Configuration), a bell (Alarm Displays), a monitor with a pulse line (DeltaV Applications), a gear (Display Tag Settings), and a camera (Custom Buttons). Below the icons, there are navigation buttons: a left arrow, a right arrow, an up arrow, and a home icon. The main content area shows a dropdown menu with 'OVERVIEW' selected, and two instances of 'PAD102_OVRVW'. On the right side, there is a 'Relocation Button' (three horizontal lines) and a 'Date/Time' display showing '04-Mar-22 2:15 PM'. The user is logged in as 'PW1DEMO.SPARTAN.DELTAV.LOCAL\ADI'. A 'Windows Format' button is also visible.

Navigation Bar Settings:

- Enable Navigation Bar
- Navigation Source: Display Hierarchy
- Highest Level to show: 1
- Number of Levels to show: 1
- Style: Single Bar
- Show Alarm Rollup: Primary Control Display
- Enable Display History Buttons
- Enable Display Relocation Button
- Show Home Button

Faceplate Contextual display

- Live has a new visual presentation for Faceplate displays.
- Similar feature set developed with GEMs
 - PV and SP bar Graph with new Color scheme
 - OUT bar moved to horizontal position
 - Embedded Trend object
- Includes familiar information
 - Module Name and Description at top
 - PV, OUT variables with Engineering Units
 - SP and OUT Slew keys
 - Mode control
 - Alarm Summary
 - Application buttons

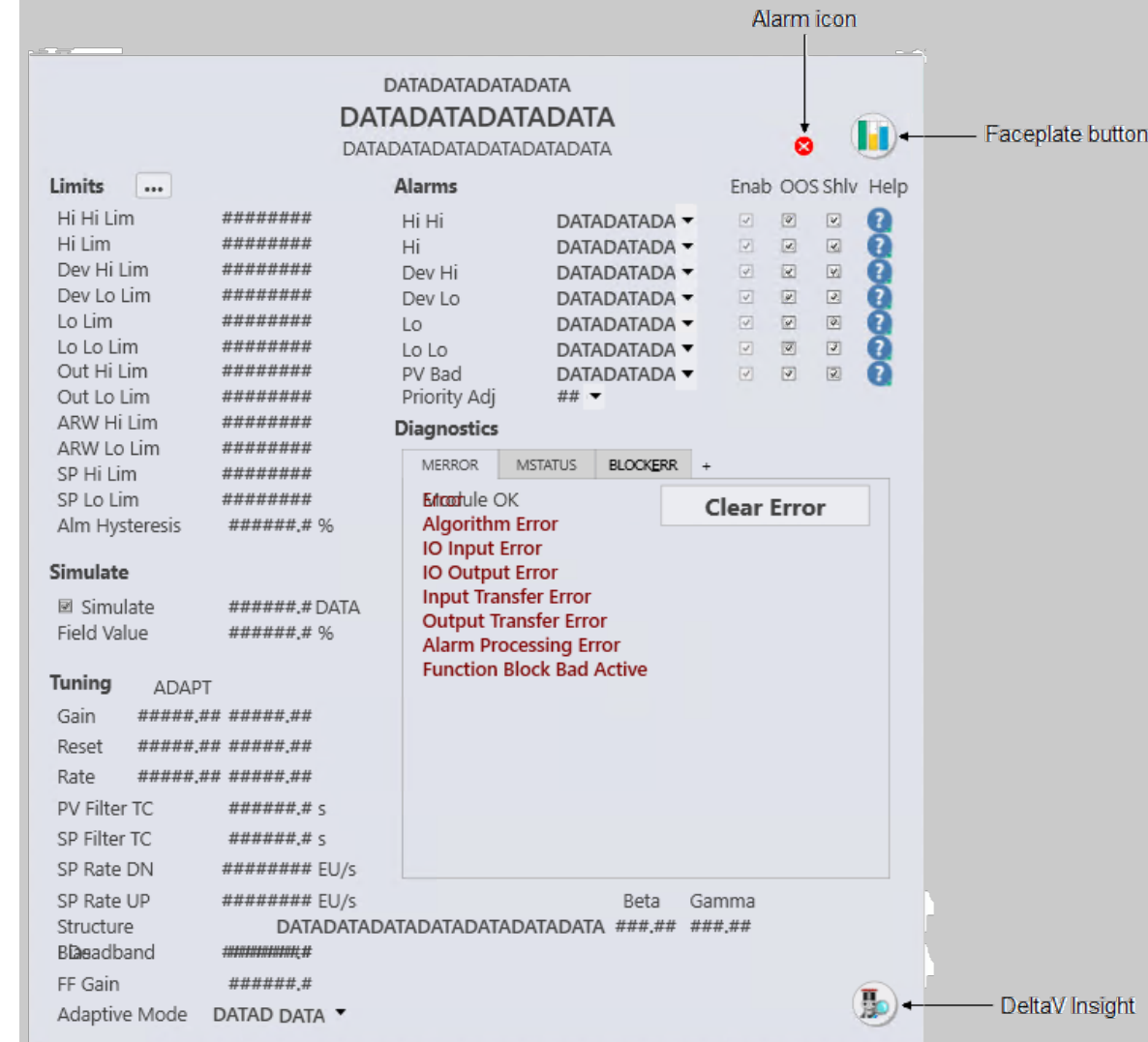


Detail Contextual display

- Live Detail Displays follow similar layout and functionality as Operate displays
- Data based on Module Primary Function Block and Alarms
 - GEMS are used to create/manage groups of items allowing for consistent Display layouts managed in one location:
 - Alarm and parameter Tuning limits
 - Alarm Configuration data
 - Module Health information
 - Primary Block Tuning variables

Each Module Class is assigned a purpose-built Faceplate and Detail contextual display defined in Graphic Studio.

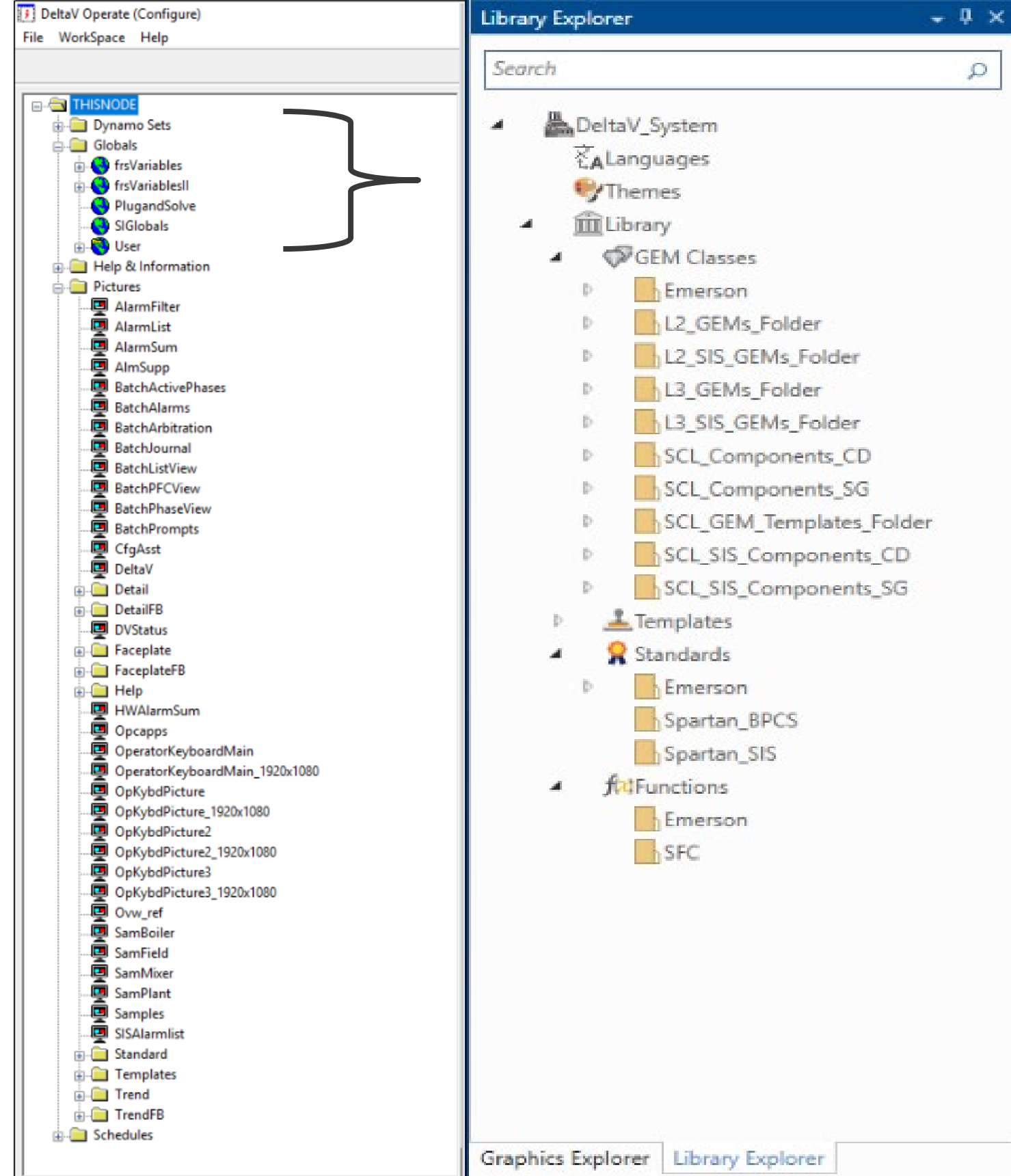
Note: L4 display names must match Operate display names to allow concurrent operation during transition.



Graphics Studio review

Library of Global Objects (HMI Toolkit)

- Implementation of the Customer HMI Philosophy
- Separate Library with expandable tree structure
 - GEMS (GraphicalEleMent)
 - Standards (Global items)
 - Functions (Color Tables etc)
- Native THEME management
- Multi Language support



Class Based GEMs

The most powerful Feature of Live is the introduction of Class based GEMs

- Instances are linked to Class object
- Class GEM can include Sub Class GEMs
- GEM Behavior can be linked to Functions
- Functions can be linked to Standards

The result is a highly flexible architecture that delivers consistent behavior through class inheritance and global objects

Note: Instances also support overrides to customize an instance while still maintaining adherence to original GEM features

DATADATADATADATA



PV: ##### DATADATA

SP: #####

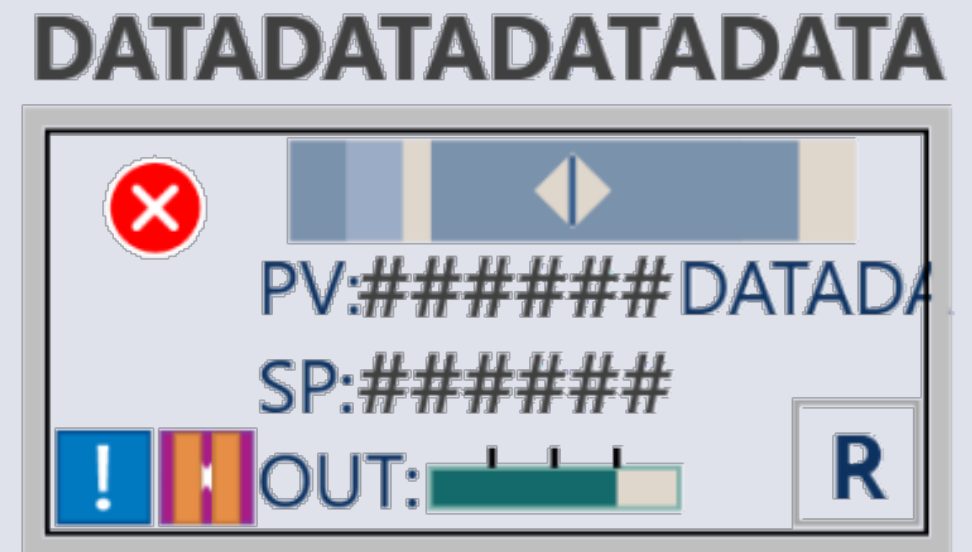
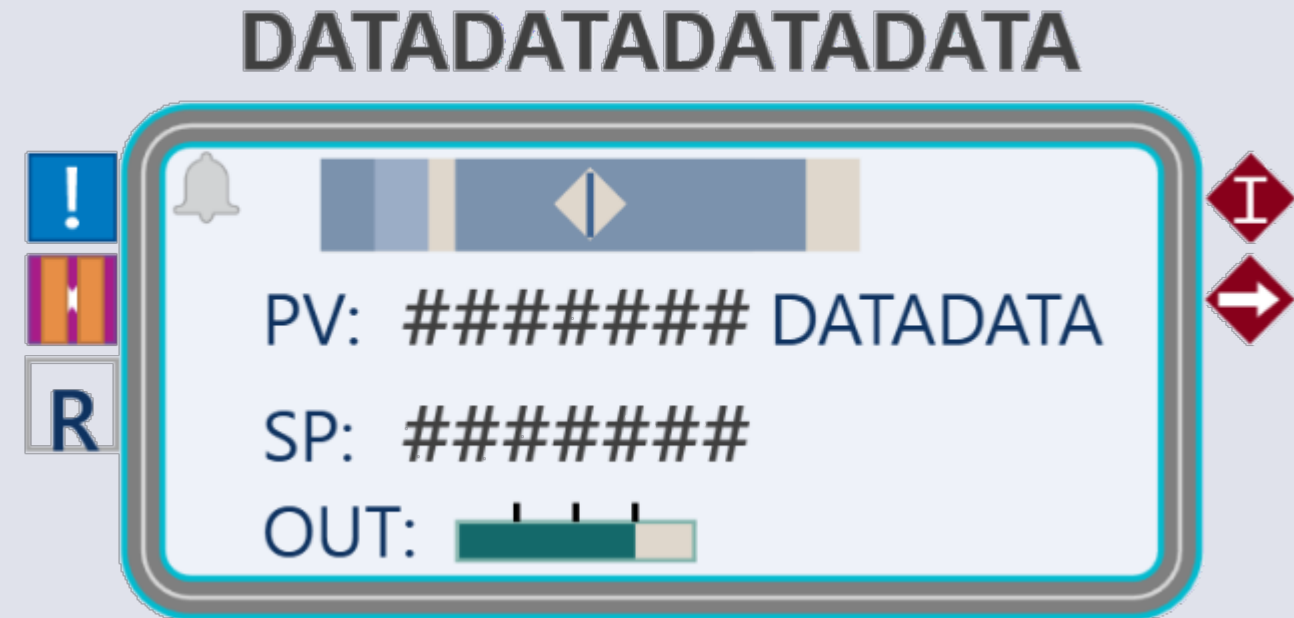
OUT: 



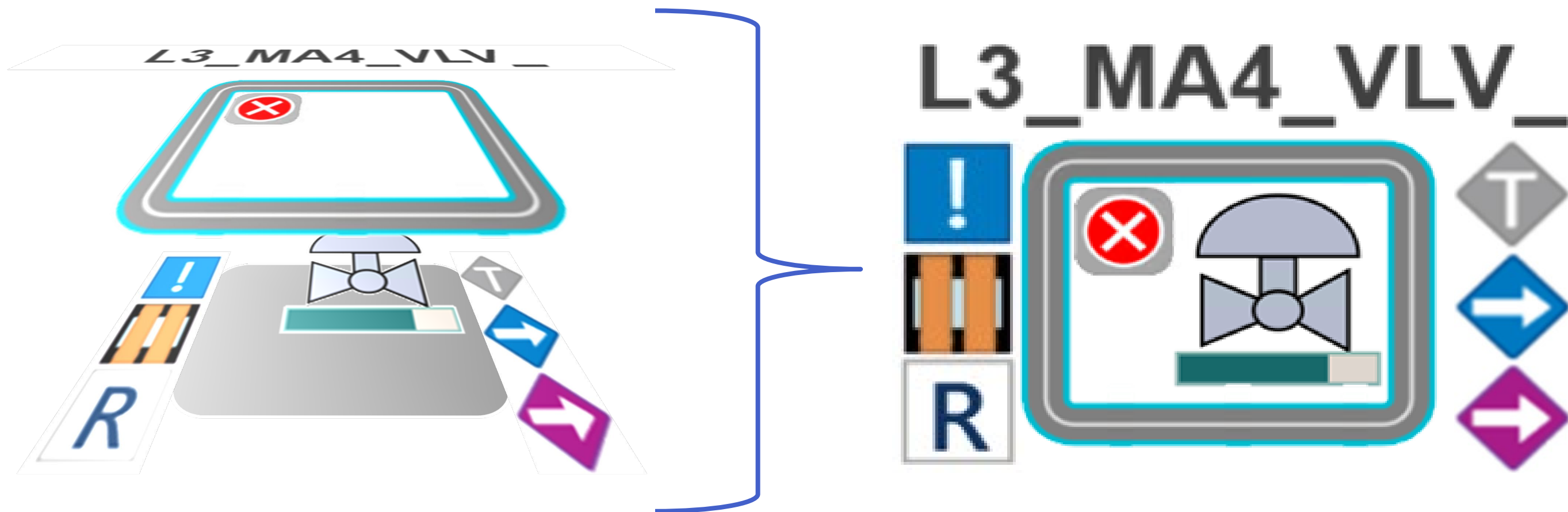
OOB GEMS vs Redesign

- Out Of Box GEMs in Live reproduce DeltaV Operate High Performance Dynamos
 - Use the same object structure
 - Inefficient design with duplicated Interaction logic
 - No new features
- This drove a redesign to deliver a design based on Live
 - Resizable Frame instead of 8 fixed sizes (Click and Drag)
 - Separate NAME GEM for ease of repositioning
 - GEM background for improved presentation in normal conditions
 - Stable frame size with condition icons external to frame.
 - A single Interaction box avoiding duplicate code.

Result is more manageable design with less runtime code and improved usability.

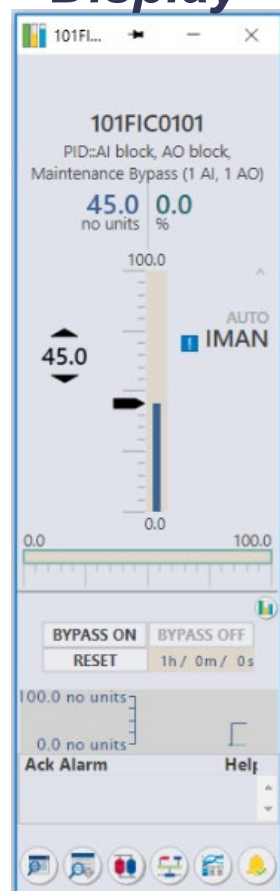


Building a Display GEM

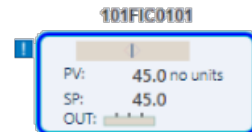


Using Library objects

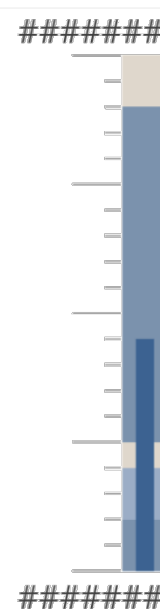
Contextual Display



L3-GEM



CD-GEMS



SUB-GEMS



Functions

F_AbnormalMode_AD_Rule

Basics

- Information**
 - Title: AbnormalMode Detection Function
 - Description: BPCS: GEM: Function to Animate AbnormalMode Icon
- Inputs**
 - Add Input

Name	Title	Type	Default Value
Tag	Module Name	String	
FB	Function Block	String	
- Calculations**
 - Conversion Table**
 - Conversion Type: Rule-Based
 - Add Rule

Rule	Output
(DLSYS.CondRead([Tag+ '/' +FB+ '/MODE.ACTUAL.STR;']) != DLSYS.CondRead([Tag+ '/' +FB+ '/MODE.NORMAL.STR;']))	True
(DLSYS.CondRead([Tag+ '/' +FB+ '/MODE.ACTUAL.STR;']) != DLSYS.CondRead([Tag+ '/' +FB+ '/MODE.TARGET.STR;']))	True
Otherwise	False

Standards

S_AlarmColor03	Blue
S_AlarmColor04	Purple
S_AlarmColor05	Purple
S_AlarmColor06	Purple
S_AlarmColor07	Purple
S_AlarmColor08	Yellow
S_AlarmColor09	Yellow
S_AlarmColor10	Yellow
S_AlarmColor11	Yellow
S_AlarmColor12	Red

Library Standards

NAME	VALUE	TITLE	DESCRIPTION	TYPE
(Spartan)				
AlarmPriority				
SK_Common				
S_AbnormalMode	!	S_AbnormalMode	Abnormal Mode Indication	Image
S_ActiveStepsText	Active Steps	Active Steps	Active Steps	MultiLanguageString
S_ActiveTransitionsText	Active Transitions	Active Transitions	Active Transitions	MultiLanguageString
S_ActualText	Actual	S_ActualText	Actual Label	MultiLanguageString
S_AlarmsText	ALARMS[1]	S_AlarmsText	Alarms Label	MultiLanguageString
# S_AnalogVlvOpen	2	S_AnalogVlvOpen	Analog Valve Open	Number
S_BadIO	X	S_BadIO	Bad IO Indication	Image
S_BypassExpirationImminent	Bypass Expiration Imminent	Bypass Expiration Imminent	Bypass Expiration Imminent	MultiLanguageString
S_BypassOff	Do you want to remove bypass on	S_BypassOff	Bypass Off	MultiLanguageString
S_BypassOn	Do you want to bypass	S_BypassOn	Bypass On	MultiLanguageString
S_DCFill0		S_DCFill0	DC Fill 0 Color	Color
S_DCFill1		S_DCFill1	DC Fill 1 Color	Color
S_DCFill2		S_DCFill2	DC Fill 2 Color	Color
S_DCFill3		S_DCFill3	DC Fill 3 Color	Color
S_DCFill4		S_DCFill4	DC Fill 4 Color	Color
S_DCFill5		S_DCFill5	DC Fill 5 Color	Color
S_DCFillElse		S_DCFillElse	DC Fill Else	Color
S_DCLine0		S_DCLine0	DC Line 0 Color	Color

System Wide standards used to define the look and feel of displays.

Library Functions

The screenshot displays the 'Library Explorer' window with a tree view on the left and a list of functions in the center. The 'Common' folder under 'Spartan' is selected. The 'F_AlarmColor' function is highlighted in the list, and its configuration panel is shown on the right.

NAME	TITLE	DESCRIPTION	TYPE	LOCKED
(Spartan)				
SK_Common				
# F_Direction				
F_MonthName				
SCL_F_GuageBarColor				
AB SCL_F_GuageFont				
SCL_F_RunColor				
SCL2_F_CASTxtStatColor				
SCL2_F_CondColor	Sequence Condition Color	BLR101 Sequen		
SCL2_F_FirstOutTripAlarm	FirstOut Trip Alarm Color	FirstOut Trip Al		
SCL2_F_FirstOutTripText	FirstOut Trip Text Color	FirstOut Trip Te		
SCL2_F_InfoTxtStatColor				
01 SCL2_F_NoPermitVisibility				
SCL2_F_SeqButton	Seq Button Color	BLR101 Sequen		
SK_F_Sim_DI_Off_ButtonColor				

F_AlarmColor

Basics

Information

Title: Alarm Color

Description: Function to Animate Alarm Color based on Priority; Version: 0

Inputs

Add Input

Name	Title	Type	Default Value
PRI	Alarm Priority Number	Float	0

Calculations

Conversion Table

Conversion Type: Value-Based

Evaluation Basis: PRI

Add value

Value	Output
-115	...larmColor15
-114	...larmColor14
-113	...larmColor13
-112	...larmColor12
-111	...larmColor11

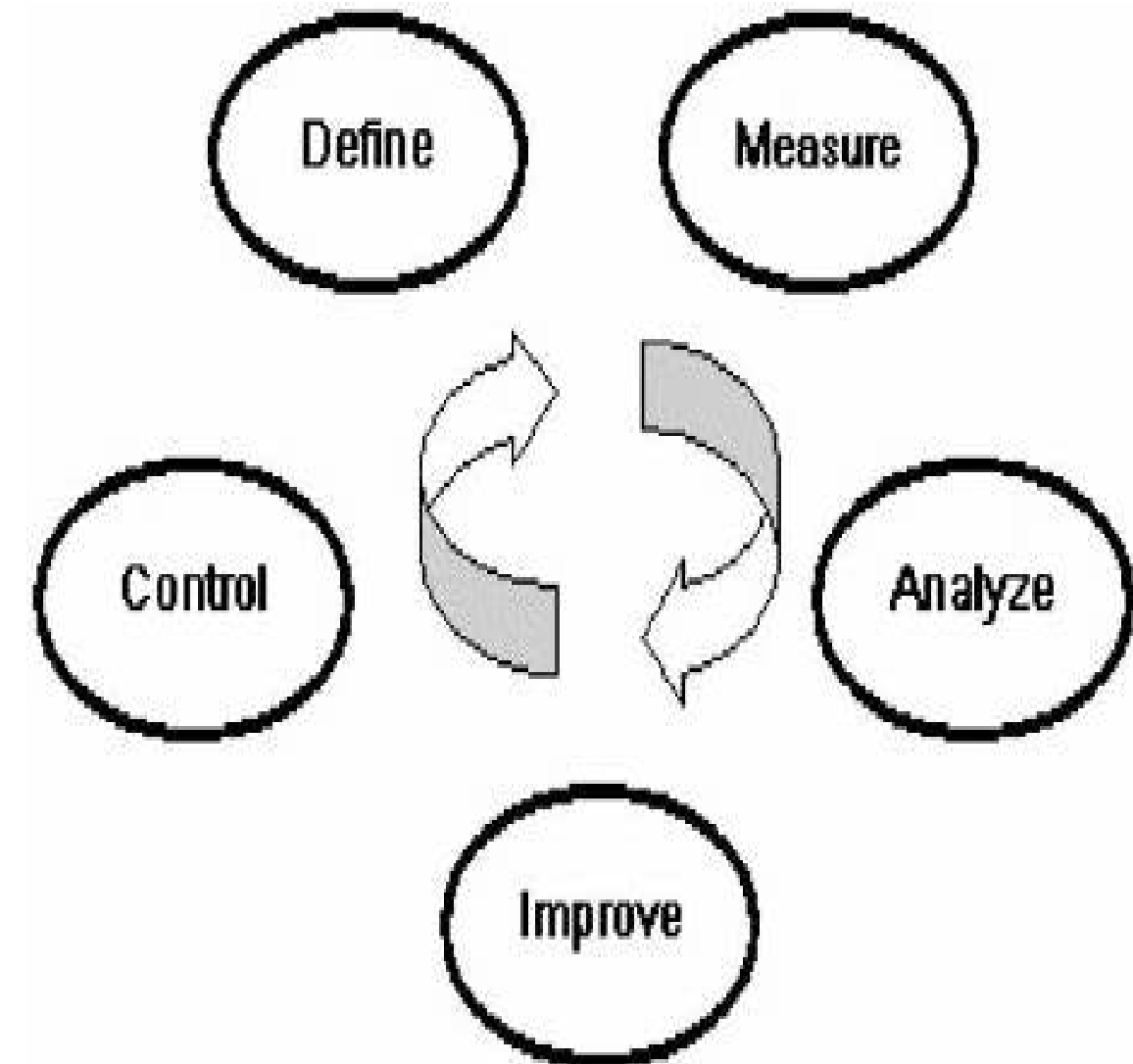
Results & Benefits

- Project Outcomes
 - Custom GEM Library
 - Standardization implementation
 - Team capability development
 - Library that can be used in other business units
- Long-Term Benefits
 - Maintainable solution
 - Enhanced team knowledge
 - Ahead of the forced migration due to obsolescence
 - Improved Operator efficiency with the better implementation of HCD principles and effective L2 displays



Lessons Learned So Far..

- Start logging hours, even though “easy” is a theme heard a lot at the conference, it is completely new, and it takes time to get your head around, from new scripting language to just “how do I do that”.
- A tool for managing the work that needs to be completed is crucial, produce what works best for you and your team, just make sure the process is not more work than the conversion.
- There are multiple ways to do the same thing, you may or may not get it right out of the gate, need to time to react and even go back and change.
- There are still some frustrations present, take a breath, get some air.



Path Forward

- Future Development
 - Continue conversion to Live
 - At some point new development in Operate will need to stop
 - Team training
 - Add to the tracking application something that helps tracks changes on Operate entities, something that flags and captures the changes other than actions
- Knowledge transfer
 - Documentation
 - Training materials
 - Best practices



Wish List

- Feature enhancements in Live
 - If Live could be decoupled from entire version solutions. Graphics and Control should be segregated
 - Feature additions in Live should be able to be installed feature additions without having to wait for full version of DeltaV to be released
 - Should be able to install “Graphics Only” on Operator Stations so that during a Control System Side upgrade, we can leave the Operators out of it
 - Primary Display, Faceplate, Detail Display need to be decoupled from Control Module. Changing name of a graphic element should not require thousands of downloads



Development Best Practices

- GEM Development Strategy Component design principles
 - Configuration guidelines
 - Use of Functions and Standards
 - Common GEM Properties
 - Approach to embedded GEM
 - Testing procedures
 - Reduce Testing scope with global items
 - Class based deployment and Functions
 - Documentation standards
 - Using Title and Description fields in Live
 - Export As Built lists to documents



Recap

Project Scope Achieved

- Major Start to HMI migration for two facilities
- Good portion of graphics running in parallel within 12-16 months

Team Achievements

- In-house development capability
- Custom tracking application
- Standardized GEM library

Success Factors

- Structured project management
- Custom tracking solution
- Team engagement and ownership
- Operations collaboration
- Operator feedback and stakeholder buy-in

Long-Term Value

- Cost-effective solution
- Enhanced team capabilities
- Maintainable solution
- Foundation for future projects

Q&A/Contact Information

- Summary of Key Points
- Contact Information
- Additional Resources





EMERSON EXCHANGE 2025

ACCELERATING
INNOVATION

Thank You