Displays Module How-to

Create, customize, and manage visual displays.

Overview

This guide walks you through creating operator interfaces with the Displays module. You'll learn to design displays, configure layouts, and deploy to multiple platforms from a single design.

Prerequisites:

- Tags configured for data display
- Understanding of UI requirements
- Basic knowledge of display types (HMI vs Dashboard)

Guide Structure

This overview links to specific guides for different aspects:

- **Creating Displays** Basic display creation and configuration
- **Drawing Canvas Displays** Process graphics and P&IDs
- **Building Dashboards** Responsive data visualization
- **Working with Symbols** Using and customizing the symbol library
- **Configuring Layouts** Multi-frame application structure

Quick Start: Your First Display

1. Create a Display

1. Navigate to **Displays Draw**
2. Click **New Document** button
3. Configure:
    - **Name**: MainDisplay
    - **Engine**: Portable (works on all platforms)
    - **Template**: Canvas (for process graphics) or Dashboard (for KPIs)
4. Click **OK**

2. Add Basic Elements

Quick Method - Drag Tags:

1. From right panel, drag tags directly to canvas
2. System auto-creates appropriate visualization

Manual Method - Add Components:

1. From left panel, select component (Button, Gauge, etc.)
2. Drag to display
3. Double-click to configure

3. Test Your Display

1. Click **Preview** button in toolbar
2. Or keep solution running and save display
3. View in Rich Client or refresh Web Client

Display Types and When to Use Them

Type	Purpose	Best For	Resize Behavior
Canvas	Process graphics	P&IDs, equipment layouts	Maintains relationships

Dashboard	Data visualization	KPIs, reports, analytics	Responsive grid
Page	Main content	Primary displays	Replaces current
Popup	Overlay content	Trends, details	Non-modal overlay
Dialog	User interaction	Confirmations, inputs	Modal, blocks UI

Platform Deployment

Engine Options

Setting	Deploys To	Use When
Portable	WPF + HTML5	Default choice, maximum compatibility
WPF Only	Windows clients	Using Windows-specific features
HTML5 Only	Web browsers	Web-only deployment

RESS Technology

For mobile devices, RESS (Responsive + Server-Side) automatically:

- Detects device type (phone/tablet/desktop)
- Delivers optimized content
- Maintains responsive behavior within each class

Basic Configuration Workflow

Step 1: Plan Your Application

1. **Define Layout Structure**
    - Header: Navigation/title
    - Menu: Main navigation
    - Content: Primary displays
    - Footer: Status/information
2. **Identify Display Types**
    - Process overviews (Canvas)
    - KPI dashboards (Dashboard)
    - Detail popups (Popup)
    - User inputs (Dialog)

Step 2: Create Layout

1. Go to **Displays Layouts**
2. Click **Add** button
3. Name: "MainLayout"
4. Map displays to regions:
    - Header HeaderDisplay
    - Menu NavigationMenu
    - Content (changes dynamically)

Step 3: Configure Startup

1. Navigate to **Displays Client Settings**
2. Set **Initial Layout**: MainLayout
3. Configure display behavior:
    - Theme selection
    - Security indicators
    - Multi-monitor (if needed)

Essential Display Elements

Common Components

Component	Use For	Key Properties
TextBox	Values, labels	Text, Font, Alignment
Button	Commands	Action, MouseDown
ComboBox	Selection	Items, SelectedValue
DataGrid	Tables	DataSource, Columns
TrendChart	Time series	Tags, TimeRange
AlarmWindow	Alarm display	Filter, Columns

Adding Dynamics

Quick dynamic properties (double-click element):

- **Visibility**: Show/hide based on tag
- **Color**: Change based on value
- **Position**: Move with tag value
- **Size**: Scale with data
- **Rotation**: Rotate based on angle

Display Navigation

Opening Displays

From Script:

csharp

```
// Replace content area
@Display.ProcessOverview.Open();

// Open as popup
@Display.TrendPopup.OpenPopup();

// Open as dialog
@Display.ConfirmDialog.OpenDialog();
```

From Button:

1. Select button
2. Set Action property
3. Choose "OpenDisplay"
4. Select target display

Changing Layouts

csharp

```
// Switch entire layout
@Client.OpenLayout("OperatorLayout");

// Just change content
@Display.NewDisplay.Open();
```

Display CodeBehind

Basic Structure

csharp

```
public void DisplayOpening()
{
    // Initialize before display opens
    LoadUserPreferences();
}

public void DisplayIsOpen()
{
    // Execute after display is visible
    StartDataRefresh();
}

public void DisplayClosing()
{
    // Cleanup when closing
    SaveUserSettings();
}
```

Handling User Input

csharp

```
public void Button1_Click(object sender, EventArgs e)
{
    @Tag.PumpStart = 1;
    @Display.PumpDetails.OpenPopup();
}
```

Common Tasks

Making Displays Responsive

1. Use **Dashboard** template
2. Set **OnResize**: Responsive
3. Configure grid columns/rows
4. Test on different screen sizes

Supporting Multiple Languages

1. Create dictionaries in **Displays Localization**
2. Use syntax: `@(LocalizationKey)`
3. Switch at runtime: `@Client.Localization = "Spanish"`

Optimizing for Mobile

1. Create mobile-specific displays
2. In Layout, configure Mobile column
3. Use larger buttons and fonts
4. Minimize text input

Performance Tips

Limit elements per display - Keep under 500 elements

Use symbols - Reuse instead of recreating

Optimize images - Compress and resize appropriately

Minimize animations - Use only necessary dynamics

Test on target platform - Verify performance early

Troubleshooting

Display Won't Open

- Check BuildStatus for errors
- Verify display name spelling
- Confirm RunSecurity permissions
- Check layout configuration

Poor Performance

- Reduce element count
- Simplify animations
- Check CodeBehind for loops
- Use async methods

Platform Differences

- Verify Engine setting
- Test on target platform
- Check component compatibility
- Review feature limitations

Displays Module Links

Explanation - to understand concepts

Modules / User Interactions / Displays Module

Tutorials - to learn by doing

Tutorials / User Interactions / Displays Module Tutorial

How-to Guides - to accomplish specific tasks

How-to Guides / User Interactions / Displays Module How-to Guide

Reference - technical details

Technical Reference / User Interactions / Displays Module Reference

---

In this section...
