Creating Dashboard Displays How-to

Build responsive operator dashboards with automatic layout adaptation

How-to Pillars Operator Displays Dashboard | Tutorial | How-To Guide | Reference

---

Overview

This guide covers creating responsive dashboards that automatically adapt from grid layout to mobile stack view, with real-time event-driven updates.

Prerequisites

- Tags configured for data display
- Understanding of KPI requirements
- Data sources for charts and gauges

Creating a Dashboard
Dashboard Architecture
Adding Dashboard Elements
Configuring Blocks
Dashboard Styling
Real-Time Updates
Mobile Optimization
Runtime Features
Common Dashboard Patterns
Best Practices Checklist

Creating a Dashboard

New Dashboard Display

1. Navigate to **Displays Draw**
2. Click **Create a New Document**
3. Configure: • **Name**: Dashboard name • **Engine**: Portable (.NET and HTML5) • **Template**: BasicDashboard
4. Click **OK**

Dashboard Architecture

Layout System

Display Width	Layout Control	Behavior
> 400px	WPF Grid	Multi-column responsive grid
400px	Vertical StackPanel	Single column mobile view

The dashboard automatically switches between Grid and StackPanel based on display width, ensuring optimal layout on all devices.

Block Controls

Control	Designer	Runtime	Function
Move	?	?	Reorder blocks
Delete	?	-	Remove block
Maximize	-	?	Full screen view
Resize	?	?	Drag gridline separators

Adding Dashboard Elements

Available Components

- **DataGrid** - Tabular data with sort/filter
- **Trend Charts** - Real-time time series
- **Pie/Bar Charts** - Data visualization
- **Gauges** - KPI indicators
- **Symbols** - Library or custom components

Container Types

Container	Purpose	Behavior
Group	Keep items together	Acts as single block
Horizontal StackPanel	Side-by-side layout	Items in row
Vertical StackPanel	Stacked layout	Items in column
WrapPanel	Flow layout	Wraps to next line

Using Groups

1. Click Group icon in toolbar
2. Drop container on dashboard
3. Place related items inside
4. Group moves/resizes as one unit

Configuring Blocks

Block Properties

Property	Default	Description
Title	-	Block header (supports tags)
Margin	4	Space around block
Padding	0	Space inside block
Width/Height	Auto	Size control
Hide on Mobile	Unchecked	Hide in stack view

Dynamic Titles

```
// Use curly brackets for real-time values
Title: {Tag.ProductionRate} units/hr
Title: {Client.DateTime}
Title: Line Status: {Tag.LineStatus}
```

Dashboard Styling

Appearance Options

- **Themes** - 12 built-in themes (light/dark)
- **Fill** - Block background color
- **Font** - Text styling

CodeBehind Customization

.NET:

```
public void OnDashboardCustomizeItem(TCell item)
{
    item.TitleElement.TextAlignment = TextAlignment.Center;
    item.TitleElement.FontStyle = FontStyles.Italic;
    item.TitleElement.FontSize = 20;
    item.TitleElement.Foreground = Brushes.Black;
}
```

HTML5:

```
this.OnDashboardCustomizeItem = function(item)
{
    item.TitleElement.FontSize = 18;
};
```

Real-Time Updates

Event-Driven Architecture

Dashboard updates are fully event-driven:

- All tags automatically subscribe to server
- Changes trigger immediate updates
- Typical latency: 100ms - 1 second
- No polling or refresh configuration needed

Benefits:

- Minimal network traffic
- Instant value updates
- Efficient resource usage

Mobile Optimization

Responsive Behavior

1. Dashboard detects display width
2. At 400px, switches to StackPanel
3. Blocks stack vertically
4. Hidden blocks don't render

Mobile Configuration

- Check "Hide Block on Mobile" for non-essential KPIs
- Test at 400px width during design
- Prioritize critical information at top

Runtime Features

Operator Controls

- **Maximize** - Full screen focus on block
- **Restore** - Return from maximized
- **Resize** - Drag gridlines between blocks
- **Reorder** - Move blocks to new positions

Common Dashboard Patterns

Operations Dashboard

```
[Group: Production Status]
  Line 1 Gauge | Line 2 Gauge | Line 3 Gauge
[Trend Chart - Full Width]
[Group: Alarms and Events]
  Active Alarms Grid | Event Log
```

KPI Dashboard

```
[Horizontal StackPanel: Metrics]
  Revenue | Costs | Efficiency | Quality
[Group: Trends]
  Production Trend | Quality Trend
[DataGrid: Detail Data]
```

Best Practices Checklist

Design

- Use groups for related items
- Test at 400px for mobile view
- Place critical KPIs at top
- Apply consistent themes

Performance

- Leverage event-driven updates
- Limit complex visualizations
- Test with production data volumes

In this section...
