Displays Layouts Reference

Organize displays into responsive layouts.

Reference **Modules** **Displays** Layouts Concept | How-to Guide | Reference

---

Displays Layouts (Reference) the Operator UI's structural framework, organizing displays into regions that persist during navigation while the Content area changes with each page.

Layout provide:

- Consistent navigation structure
- Platform-adaptive regions
- Header/Menu precedence control
- Automatic display fallback
- Multi-platform optimization

Layout Workspace
Configuration Workflow
Platform Adaptation Examples
Panel Regions
Layout Design Considerations
Layouts in Runtime
Best Practices Checklist
Troubleshooting

Layout Workspace

The Layout configuration interface has four main areas:

1. **Layout List**
2. **Panel Configuration**
3. **Page Assignment Table**
4. **Platform Previews**

(1) Layout List

Search, +, ×, sort

Startup
Dashboard
Layout1
Layout2
...

(2) Panel Configuration

Panel Regions

<table>
<tr><td colspan="2">Header</td><td>Header Edit: Stretch / Left / Center / Right</td></tr>
<tr><td rowspan="2">Menu</td><td>Content Submenu</td><td rowspan="2">Footer Edit: Stretch / Left / Center / Right</td></tr>
<tr><td>Footer</td></tr>
</table>

(3) Page Assignment Table

Region	Page	Mobile	Tablet	Description
Header	Header			Display for the Header panel
Footer				Display for the Footer panel
Menu				Display for the Menu panel
Submenu				Display for the Submenu panel
Content	MainPage			Display for the Content panel

(4) Platform Previews

Desktop Web Preview

Preview area

Desktop Windows WPF

Preview area

Mobile Preview

Preview area

Tablet Preview

Preview area

Configuration Workflow

Step 1: Manage Layouts

Action	Method	Result
Add Layout	Click + button	Create new layout
Delete Layout	Select and click ×	Remove selected

Select Layout	Click layout name	Load for editing
Search	Use search box	Filter list
Sort	Click sort button	Reorder list

Step 2: Configure Selected Layout

2.1 Organize Regions

Configure which panels are active and their behavior:

Region	Enable/Disable	Alignment Options
Header	Always enabled	Stretch/Left/Center/Right
Menu	Optional	Fixed left position
Content	Always enabled	Auto-size center
Submenu	Optional	Fixed right position
Footer	Optional	Stretch/Left/Center/Right

2.2 Define Pages

Assign pages to each active region:

Assignment Rules	Behavior
Page column filled	Display assigned page
Mobile/Tablet empty	Inherit from Page column
Mobile/Tablet with name	Platform-specific override
Any invalid name	Hide region on that platform

2.3 Preview Results

Check layout appearance across platforms:

- Desktop Web (HTML5/WebAssembly)
- Desktop Windows (WPF)
- Mobile (Portrait phone)
- Tablet (Landscape/larger screen)

Platform Adaptation Examples

Desktop Configuration	Mobile Result	Tablet Result
All 5 regions active	Header + Content only	Header + Menu + Content
Complex menu	Hamburger menu	Simplified menu
Footer with status	Hidden footer	Minimal footer
Wide submenu	Hidden	Narrow tools panel

Panel Regions

Region	Purpose	Position	Dynamic
Header	Logo, title, main navigation	Top	No

Menu	Primary navigation links	Left	Optional splitter
Content	Main display area	Center	Yes
Submenu	Secondary navigation	Right	Optional
Footer	Copyright, links, info	Bottom	No

Regions Details

Header

- **Position**: Top of application
- **Purpose**: Branding, main navigation, user info
- **Alignment**: Controls interaction with Menu/Submenu
    - Stretch: Full width above side panels
    - Center: Centered with side panels extending up
    - Left/Right: Aligned with Menu/Submenu priority

Menu

- **Position**: Left side
- **Purpose**: Primary navigation
- **Options**: Can include resizer for width adjustment
- **Mobile**: Often becomes hamburger menu

Content

- **Position**: Center area
- **Purpose**: Main workspace that changes on navigation
- **Behavior**: Only region that switches pages during runtime

Submenu

- **Position**: Right side
- **Purpose**: Secondary navigation, tools, properties
- **Mobile**: Often hidden or moved to bottom

Footer

- **Position**: Bottom of application
- **Purpose**: Status bar, copyright, auxiliary info
- **Mobile**: Frequently hidden to save space

Layout Design Considerations

Common Layout Patterns

Standard Application

- **Header**: Navigation Bar
- **Menu**: Equipment List (with splitter)
- **Content**: Main Display (changes)
- **Footer**: Status Bar

Kiosk Mode

- **Header**: Company Logo (center)
- **Content**: Full Screen Display
- **Footer**: Hidden (_)

Dashboard

- **Header**: Title Only
- **Menu**: `Hidden (_)`
- **Content**: Dashboard Display
- **Submenu**: Quick Actions
- **Footer**: Timestamp

Customization Example

Header/Footer Alignment

Option	Behavior	Use Case
Stretch	Full width	Standard header
Left	Align left	Logo placement

Center	Center content	Title focus
Right	Align right	User controls

Menu Configuration

- Optional splitter for resizing
- Adjusts to header/footer
- Can be hidden on mobile

Mobile Considerations

Responsive Design

- Simplify navigation for touch
- Hide non-essential regions
- Optimize for portrait orientation
- Test gesture navigation

Region Removal

```
Desktop Menu: NavigationPanel

Mobile Menu: _ (removed) or replaced
```

Layouts in Runtime

Startup Layout

Set default at **Displays Client Settings**:

```
Initial Execution Conditions
  ??? Layout: [Select Layout]
```

Runtime Changes

Change entire layout:

csharp

```
@Client.OpenLayout("AlternateLayout");
```

Change content only:

csharp

```
@Client.OpenDisplay("NewDisplay");
// Or
@Display.NewDisplay.Open();
```

Layout Switching Scenarios

User Role Based

csharp

```
public void SetLayoutByRole()

{
    string role = @Security.CurrentUser.Role;
    string layout = role switch
    {
        "Operator" => "OperatorLayout",
        "Engineer" => "EngineeringLayout",
        "Manager" => "ManagerLayout",
        _ => "DefaultLayout"
    };
    @Client.OpenLayout(layout);
}
```

Time Based

csharp

```
public void SetDayNightLayout()

{
    bool isNight = DateTime.Now.Hour >= 18 ||
                   DateTime.Now.Hour < 6;
    @Client.OpenLayout(isNight ? "NightLayout" : "DayLayout");
}
```

Best Practices Checklist

- [ ] **Start Simple** - Begin with Header + Content, add regions as needed
- [ ] **Mobile First** - Design for smallest screen, enhance for larger
- [ ] **Name Clearly** - Use descriptive names for layouts and pages
- [ ]
- [ ] **Keep Headers Simple** - Essential navigation only
- [ ]
- [ ] **Use Splitters Wisely** - Only where needed
- [ ]
- [ ] **Document Purpose** - Fill description fields for maintenance
- [ ]
- [ ] **Test All Platforms** - Use preview panels before deployment
- [ ]
- [ ] **Version Control** - Track layout changes

Troubleshooting

Layout not loading:

- Verify layout exists
- Check display assignments
- Review platform compatibility
- Test startup settings

Region not showing:

- Confirm display mapped
- Check visibility settings
- Verify display engine
- Test without underscore

Mobile layout issues:

- Review mobile mappings
- Check responsive settings
- Test orientation changes
- Verify touch targets

In this section...
