Displays Client Settings Reference

Configure client display settings.

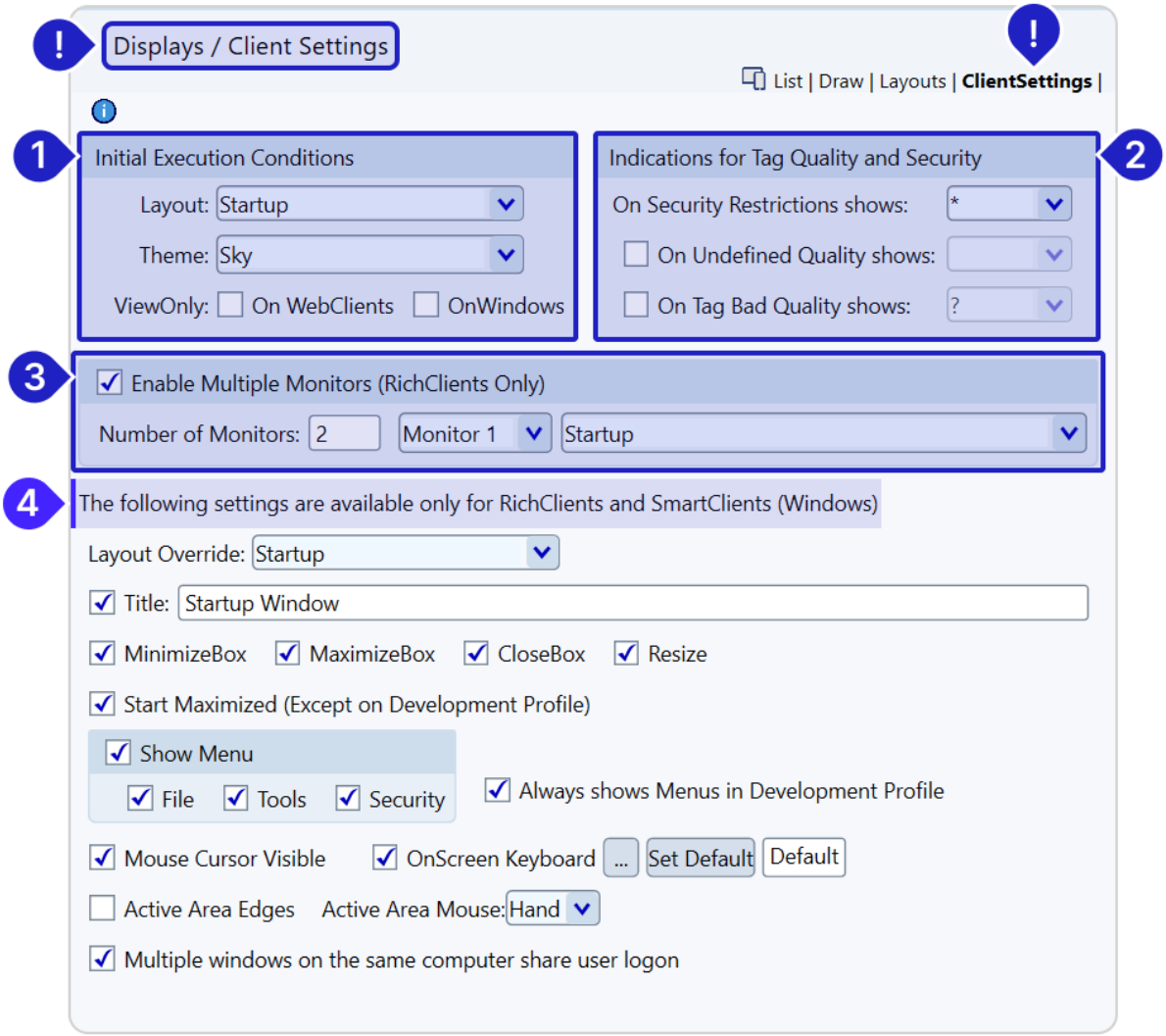
Displays Client Settings (Reference) configure the startup behavior, visual indicators, and window properties for runtime display clients across different platforms.

Client Settings provide:

- Initial layout and theme selection
- Tag quality indicators
- Security visualizations
- Multi-monitor support
- Window control options
- Platform-specific settings

Configure at **Displays** **Client Settings** to control runtime appearance and behavior.

Initial Execution Conditions
Tag Quality and Security Indicators
Multi-Monitor Configuration (RichClients)
Windows Client Settings
Special Features
Title Bar Configuration
Multi-Window Support
Runtime Properties
Best Practices
Platform-Specific Settings
Troubleshooting



Initial Execution Conditions

Setting	Description	Options
Layout	Startup layout	From Displays/Layouts
Theme	Initial theme	From Displays/Themes
ViewOnly	Platform restrictions	WebClients, Windows

Platform Restrictions

- **On WebClients** - Restrict to web browsers only
- **On Windows** - Restrict to desktop clients only

Tag Quality and Security Indicators

Visual feedback for data integrity and access control.

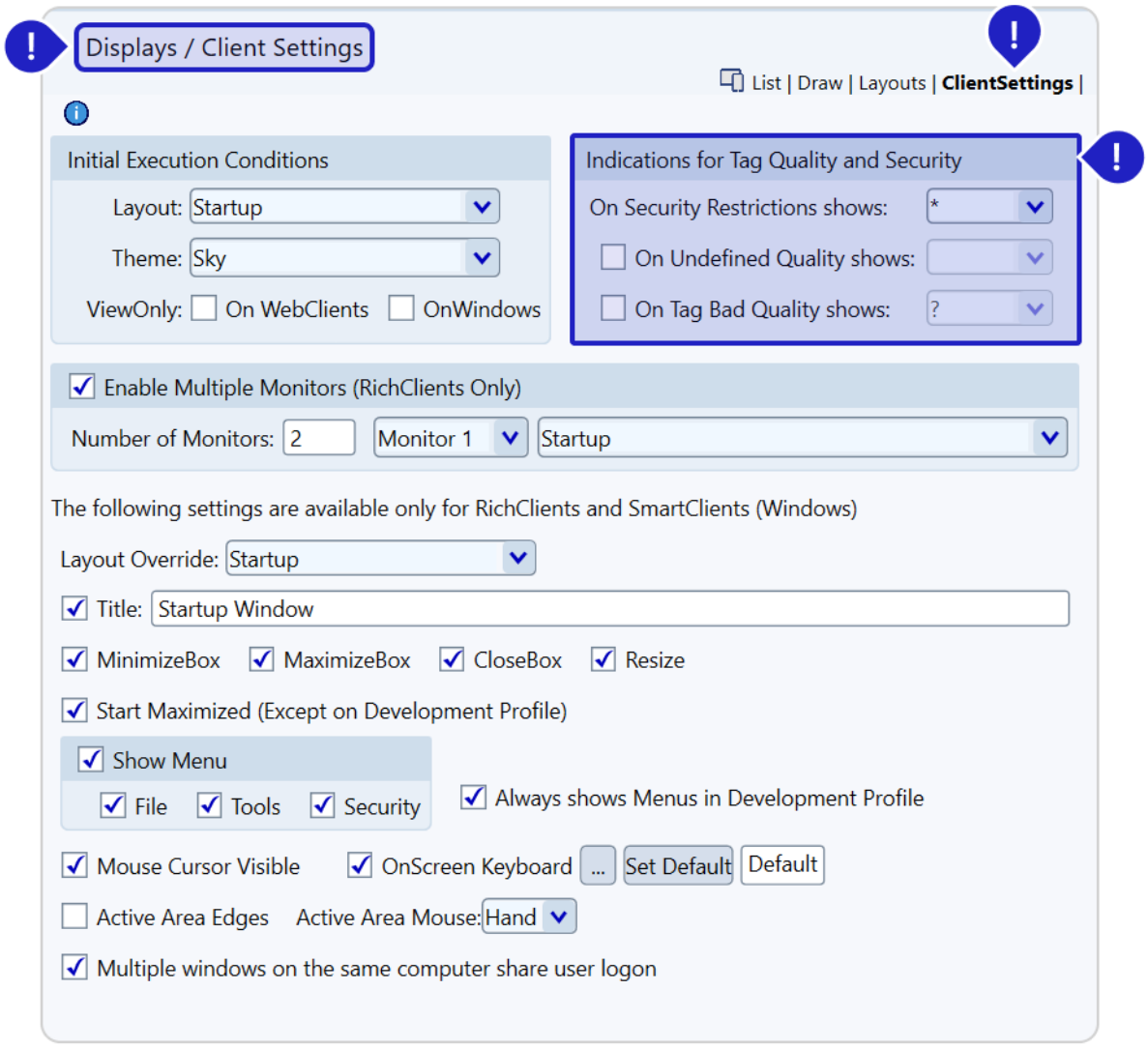
Security Restrictions

Display symbols when access denied:

- **V** - View only
- **?** - Unknown security
- **$** - Premium feature
- ***** - Restricted
- **#** - Protected
- **Strikethrough** - Disabled

Quality Indicators

Setting	Shows When	Visual
Undefined Quality	Quality unknown	? symbol
Bad Quality	Communication error	X or strikethrough



Multi-Monitor Configuration (RichClients)

Enable Multiple Displays

```
? Enable Multiple Monitors
Number of Monitors: 4
```

Monitor Assignment

Monitor	Layout	Use Case
1	Overview	Main display
2	Alarms	Alert monitoring
3	Trends	Data analysis
4	Details	Equipment focus

Windows Client Settings

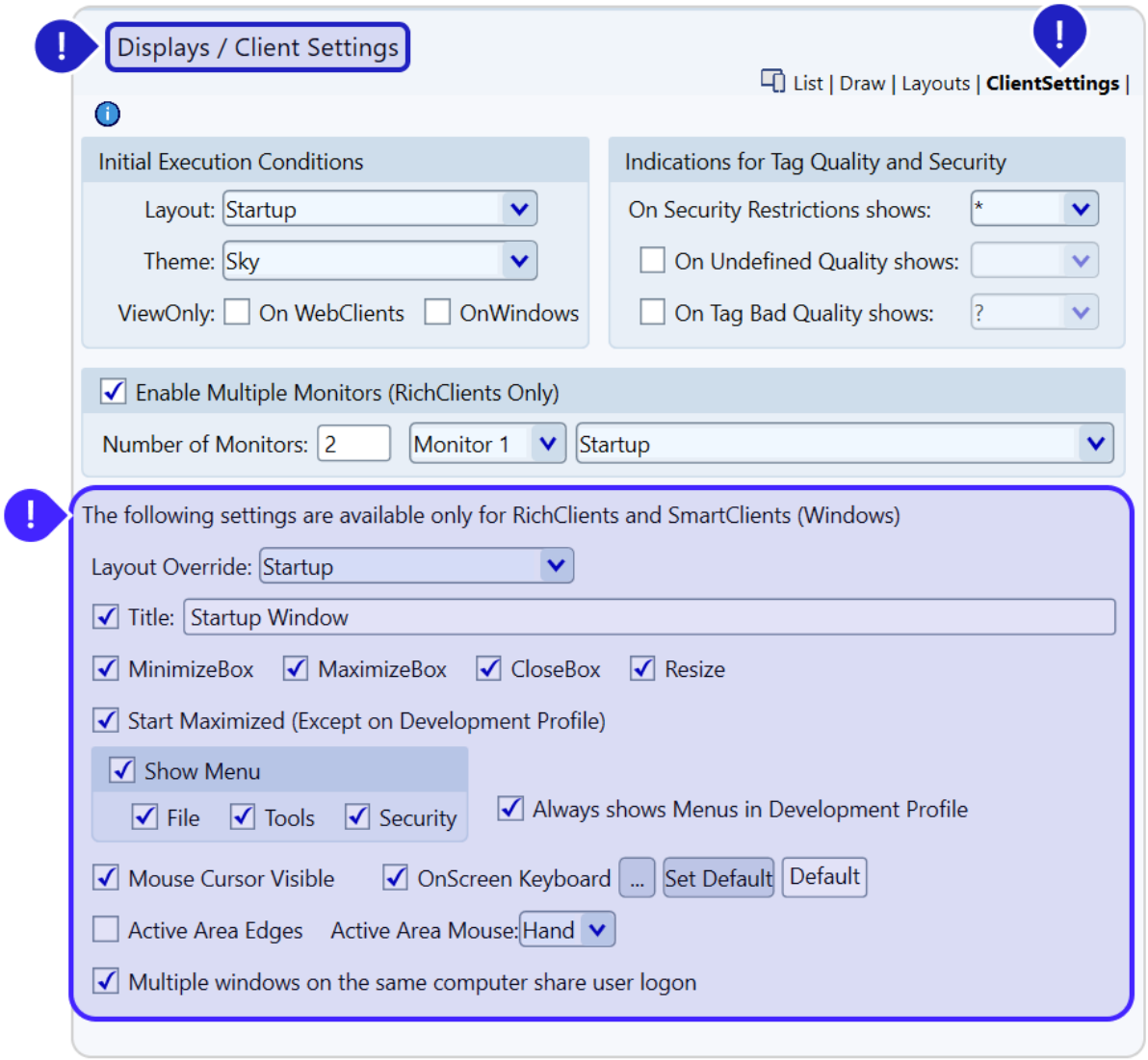
Window Controls

Control	Description	Default
MinimizeBox	Allow minimize	Yes
MaximizeBox	Allow maximize	Yes
CloseBox	Allow close	Yes
Resize	Allow resizing	Yes
Start Maximized	Full screen startup	No

Menu Visibility

Menu	Contains	Show
File	Open, Save, Exit	Optional
Tools	Settings, Options	Optional
Security	Login, Logout	Optional

Always visible in Development profile.



Special Features

Mouse Cursor

Control cursor visibility and type:

- **Visible** - Normal operation
- **Hidden** - Kiosk mode
- **Active Area** - Edge detection

Active Area Mouse Types

- **None** - Default system cursor
- **Arrow** - Standard pointer
- **Hand** - Interactive indicator

On-Screen Keyboard

> ⚠ **Rich Client / Smart Client only.** The FrameworX On-Screen Keyboard is available only on the Windows desktop clients (Rich Client and Smart Client). It is not available on Web (HTML5) or Mobile clients — on those platforms, text entry uses the device's native on-screen keyboard, provided automatically by the browser or operating system.

Enable for touchscreen operation:

csharp

```
// Show keyboard with password masking
ShowOnScreenKeyboard(
    objectName: "PasswordField",
    isPassword: true,
    callback: (ok, tag, oldVal, newVal) => {
        if (ok) ProcessLogin(newVal);
    }
);
```

To turn the On-Screen Keyboard on or off from a script, set **@Client.OnScreenKeyboard**. The property starts from the On-Screen Keyboard option in Client Settings, and a change applies at once on the Rich Client and Smart Client.

csharp

```
// Turn the On-Screen Keyboard on for this client
@Client.OnScreenKeyboard = true;
```

Title Bar Configuration

Custom Title

csharp

```
// Set runtime window title
@Client.Title = $"Plant Monitor - {DateTime.Now:yyyy-MM-dd}";
```

Title Options

- Enable/disable title bar
- Custom text
- Dynamic updates
- Include user/station info

Multi-Window Support

Allow multiple windows per user:

```
? Multiple windows on the same computer
```

Benefits:

- Single login for all windows
- Shared session state
- Independent displays
- Synchronized data

Runtime Properties

License Information

csharp

```
// Check allowed clients
int maxClients = @Info.License.AllowedRichClients;

// Monitor active sessions
int activeClients = @Server.TotalClients;

// Available slots
int available = maxClients - activeClients;
```

Best Practices

1. **Set Appropriate Defaults** - Match typical use case
2. **Configure Quality Indicators** - Clear visual feedback
3. **Plan Multi-Monitor** - Logical layout distribution
4. **Test Platform Restrictions** - Verify access control
5. **Document Settings** - Explain configuration choices
6. **Consider Touch Operation** - Enable keyboard if needed
7. **Secure Kiosk Mode** - Disable window controls

Platform-Specific Settings

Web Clients

- No multi-monitor support
- Limited window control
- Browser-based restrictions

Windows Clients

- Full multi-monitor
- Complete window control
- Native keyboard support

Mobile Clients

- Single display only
- Touch optimized
- Auto-rotation support

Troubleshooting

Layout not loading:

- Verify layout exists
- Check platform compatibility
- Review ViewOnly settings
- Test default layout

Quality indicators missing:

- Enable indicators in settings
- Check tag quality values
- Verify communication status
- Review symbol configuration

Multi-monitor issues:

- Confirm monitor count
- Check graphics drivers
- Verify layout assignments
- Test individual monitors

Keyboard not showing:

- Enable OnScreen Keyboard
- Check touch support
- Verify focus handling
- Test input methods

In this section...
