Security Permissions Reference

Assign permissions to users and groups.

Security Permissions (Reference) define access control groups that determine what users can edit in the Designer and execute during runtime. Permission groups provide:

- Module-level access control
- Designer editing restrictions
- Runtime operation permissions
- Hierarchical security levels
- Role-based authorization

Permissions are assigned to users through group membership, allowing granular control over solution access.

Pre-Defined Permission Groups
Configuration Properties
Edit Permissions
Run Permissions
Configuring Permissions
Permission Inheritance
Runtime Permission Checks
Best Practices
Common Permission Sets
Troubleshooting

Pre-Defined Permission Groups

Seven standard groups are configured by default:

Group	Typical Use	Default Permissions
Administrator	Full system control	Unrestricted access
Guest	Anonymous access	View-only, minimal rights
User	Basic authenticated access	Standard operations
Engineering	Solution development	Edit modules, test
Supervisor	Operations oversight	Monitor, reports, alarms
Maintenance	System upkeep	Diagnostics, tag values
Operator	Daily operations	Displays, acknowledge alarms

Configuration Properties

Property	Description	Required
Name	Unique group identifier. See Permission Group Names below.	Yes
Edit	Designer editing permissions	Yes
Run	Runtime execution permissions	Yes
Level	Hierarchical tier (0-255)	No
Category	Group classification	No
Description	Documentation text	No

Permission Group Names

From FrameworX 10.1.6, a permission group name starts with a letter and contains only letters, digits and underscores. Security Permissions checks the name as you type it. A name with a space or a character such as `- ! \ / .` is refused in the grid, with the reason shown. The check applies when you create or rename a group. Names already stored in a solution are kept, and the solution still builds.

Identity providers and Windows groups often use spaces in group names, for example `Domain Admins`. From FrameworX 10.1.6, such a group also matches a permission named with underscores in place of the spaces, `Domain_Admins`. The name exactly as configured is tried first. This applies to Windows Authentication and OpenID Connect sign-ins. See Security Identity Providers Reference.

Changed in FrameworX 10.1.6

In 10.1.5, Designer stored a permission group name with a space, and the solution then failed to build. From 10.1.6, Designer refuses such a name when you type it. To map an identity-provider or Windows group whose name has spaces, name the permission with underscores in place of the spaces.

Edit Permissions

Controls access to Designer modules:

Permission	Description	Affects
Unrestricted	All editing rights	Complete Designer access
EditTags	Modify existing tags	UNS tag properties
CreateTags	Add new tags	UNS structure
Security	User management	Users, permissions, policies
Scripts	Code editing	Tasks, classes, expressions
Datasets	Database configuration	Queries, tables, connections
Displays	Screen development	Pages, popups, symbols
Reports	Report design	Forms, WebData
Historian	Data logging setup	Tables, triggers
Alarms	Alarm configuration	Items, groups, areas
Devices	Communication setup	Channels, nodes, points
Startup	Runtime configuration	Execution settings
Publish	Deploy solutions	Build and distribute
Settings	Solution properties	Global configuration
Notes	Documentation	Solution notes

Run Permissions

Controls runtime operations:

Permission	Description	Impact
Unrestricted	All runtime rights	Complete control
Test	Execute test mode	Debug capabilities
Startup	Start server modules	Scripts, datasets, devices
Shutdown	Stop application	Terminate runtime
ClientStart	Start client modules	Displays, local devices
ClientShutdown	Stop client	Close displays
StartTools	Launch diagnostics	PropertyWatch, TraceWindow
ToolsSetValues	Modify via tools	Write tag values
CreateUsers	Add runtime users	Dynamic user creation
SwitchApplication	Change context	Alt-Tab, taskbar access
WebAccess	Web client login	HTML5 display access

Configuring Permissions

Access Requirements

1. Login as Administrator
2. Navigate to **Security Permissions**
3. Administrator password required for changes

Setting Group Permissions

1. Select permission group row
2. Configure Edit permissions:
    - Check modules user can modify
    - Uncheck restricted areas
3. Configure Run permissions:
    - Enable allowed operations
    - Disable restricted functions
4. Save changes

Permission Inheritance

Users inherit combined permissions from all assigned groups:

```
User: John
Groups: Operator, Maintenance
Result: Union of both group permissions
```

Example combinations:

- Operator + Maintenance = Displays + Diagnostics
- Engineering + Supervisor = Development + Monitoring
- User + WebAccess = Basic rights + Web client

Runtime Permission Checks

Checking Current Permissions

csharp

```csharp
// Current user's groups
string permissions = @Client.Permissions;

// Check specific permission
bool canEdit = @Security.HasPermission("EditDisplays");
bool canShutdown = @Security.HasPermission("Shutdown");

// Check multiple permissions
bool isAdmin = @Client.Permissions.Contains("Administrator");
```

Conditional UI Elements

csharp

```csharp
// Show/hide based on permissions
if (@Security.HasPermission("StartTools"))
{
    btnDiagnostics.Visible = true;
}

// Enable/disable functions
btnShutdown.Enabled = @Security.HasPermission("Shutdown");
```

Best Practices

1. **Principle of least privilege** - Grant minimum required permissions
2. **Use groups not individuals** - Manage through group membership
3. **Document group purposes** - Clear role descriptions
4. **Regular audits** - Review permission assignments
5. **Test permission sets** - Verify restrictions work
6. **Separate development/operations** - Different groups for each

7. **Protect Administrator** - Limit admin group membership

Common Permission Sets

Operator Standard

- Edit: None
- Run: ClientStart, WebAccess
- Use: Daily operations

Maintenance Technician

- Edit: EditTags
- Run: StartTools, ToolsSetValues
- Use: Troubleshooting

Shift Supervisor

- Edit: Alarms, Reports
- Run: Unrestricted except Shutdown
- Use: Operations management

System Engineer

- Edit: Unrestricted except Security
- Run: Test, StartTools
- Use: Development and testing

Troubleshooting

Cannot edit module:

- Check Edit permissions
- Verify group membership
- Confirm logged in correctly
- Not using Guest account

Runtime function disabled:

- Review Run permissions
- Check user's groups
- Verify permission spelling
- Test with Administrator

Permission not working:

- Clear permission cache
- Restart runtime
- Check group assignment
- Review permission conflicts

In this section...
