iX Developer 3.0
User Guide

MAEN433A
2025-06-16

Beijer ELECTRONICS

Table of Contents

1. About

1.1. About This User Guide

This user guide provides comprehensive instructions for the **iX Developer** software from **Beijer Electronics**.

IMPORTANT
This user guide applies to iX Developer version **3.x**.

TIP
To access this user guide from iX Developer, press **F1**.

1.1.1. Admonitions Used in This Manual

Admonitions call attention to critical information. Interpret them as follows:

CAUTION
Alerts you to potential hazards that may damage equipment, cause data loss, or result in minor injuries.

IMPORTANT
Highlights key information.

NOTE
Points out relevant facts and conditions.

TIP
Provides useful, non-essential information to assist you.

1.2. About iX Developer

iX Developer is an advanced software platform used for designing human-machine interface solutions on HMI panels from Beijer Electronics.

iX Developer is installed on a **development PC**, where projects are developed and designed. The project is then compiled as an **App**, transferred to an **HMI panel** with the **OS3** operating system, to observe and control a **controller**, or a group of controllers.

Key features

- **User-friendly interface** – Drag-and-drop tools simplify the creation of graphical user interfaces.
- **Advanced functionality** – Includes built-in scripting, alarm handling, data logging, and support for multiple industrial communication protocols.
- **Customizability** – Offers extensive libraries of pre-made components and supports custom designs.
- **Cross-platform compatibility** – Integration with Beijer Electronics' HMI panels and third-party devices.

1.2.1. New Features in iX Developer 3

OS3	The most significant update in **iX Developer 3** is the **OS3** operating system. This modular architecture enhances flexibility and ensures compatibility with modern operating system standards. For detailed information about OS3, see the OS3 User Guide.
Enhanced security	To strengthen security, we have reviewed all third-party libraries, drivers, and other components to meet strict security standards. For details, see the Security chapter.
Active Directory	**Microsoft Entra** is a cloud-based identity management service. **Active Directory (Azure AD)**, part of Entra, manages user accounts and permissions to ensure secure access to applications and data. Key features for Active Directory in iX Developer are: • **Single sign-on (SSO)** – Users sign in once to access all applications. • **Multi-factor authentication (MFA)** – Adds an extra layer of security beyond passwords. • **Conditional access** – Controls access based on device, location, or other factors. For details, see the User Management chapter.
Secure project transfer	To protect project data, **iX Developer 3** now supports encrypted project transfer. A secure handshake between iX Developer and the HMI panel requires authentication with a username and password before a project can be transferred. For details, see Transfer, Fetch, Export and Import.
NuGet packages	NuGet is a package manager for .NET that simplifies adding, managing, and updating libraries in your projects. In **iX Developer 3**, NuGet replaces the old **Reference Assemblies** setup, offering several benefits: • **Easier dependency management** – Add, update, or remove libraries without manual file handling. • **Access to more libraries** – Use prebuilt solutions from Microsoft and the open-source community. • **Improved version control** – Update or revert libraries to maintain compatibility. • **Simplified updates** – Quickly apply security patches and bug fixes. • **Better organization** – Dependencies are automatically tracked for easier project maintenance. For details, see the NuGet Packages chapter.

1.2.2. Transition from iX2 to iX3

To transition from iX Developer 2 to iX Developer 3 you need an X3 HMI panel. To ensure a smooth and secure transition follow the X2 to X3 – Transition Guide.

1.3. About Targets

In **iX Developer**, a **target** is the hardware device that runs the **iX App**. Typically, this is a **Beijer Electronics HMI panel**, but can also be a PC used for simulation.

1.3.1. About HMI Panels

Apps created in **iX Developer 3.0** and later are supported on all **Beijer Electronics HMI panels** from the **X3** series, except for the **X3 web variants**.

TIP
For information about the X3 HMI panels, visit the Beijer Electronics Documentation Portal.

1.3.2. About OS3

OS3 is the operating system for all **X3 HMI panels**, excluding the **X3 web variants**. It allows the HMI panel administrator to efficiently install and manage applications, configure system settings, and ensure the end-user experience remains securely within the HMI environment.

TIP
For detailed information about OS3, see the OS3 User Guide.

1.3.3. About Apps

In iX Developer, a project is compiled as an **App**, which is then transferred to the target device.

1.4. About Controllers

iX Developer HMI panels can be connected to various types of automation equipment, such as PLCs, servos, and drives. In this manual, the term controller refers to any connected equipment.

TIP
For detailed information on specific controllers, refer to the Driver Manual for the corresponding controller.

1.5. Recommended System Specifications for iX Developer

To ensure optimal performance when running iX Developer, use a system that meets or exceeds the specifications below.

Parameter	Recommendation
RAM	8-16 GB (use 16 GB if **Visual Studio** is your code editor).
Processor	Intel 12th Gen / AMD Ryzen 6000-series or newer. Must be compatible with Windows 11. 64-bit (x64) processor required. 32-bit (x86) systems are not supported.
Operating system	Microsoft Windows 10 Home, Pro and Enterprise (22H2). Microsoft Windows 11 Home, Pro and Enterprise (24H2, 23H2).  **NOTE** Microsoft support for Windows 10 ends on 2025-10-14.
Display	Full HD resolution or higher.

Additional recommendations

- Avoid low-power CPUs and low-power modes of Windows.
- For best performance, use a workstation-class PC.
- If using a laptop, keep it connected to power.
- Performance may vary depending on background applications. Close unnecessary programs such as web browsers and resource-intensive tools.

2. Get Started

This section describe how to get started with **iX Developer**.

2.1. Install iX Developer

To install **iX Developer 3.0** on your development PC and transfer a project from version 2.53, see the X2 to X3 – Transition Guide for iX Developer.

2.1.1. Install Multiple Versions of iX Developer

You can install multiple versions of iX Developer on a single development PC. For example, you can install iX Developer 2.53 and 3.0 on the same PC.

NOTE
Copy-paste between two versions of iX Developer is **not** supported.

2.2. Launch iX Developer

Launch **iX Developer** from the **Start menu**.

2.2.1. Product Registration

When iX Developer, is launched for the first time, a registration dialog box will appear. Enter the registration key to unlock full access to all features and software updates.

2.2.1.1. Trial Version

Without registration key, you can evaluate iX Developer with full functionality for **30 days**. After the trial period, the software will remain functional, but certain features in the Project Ribbon Tab will be disabled. The 30-day trial period can not be extended.

2.2.2. Import Settings from a Previous Version of iX Developer

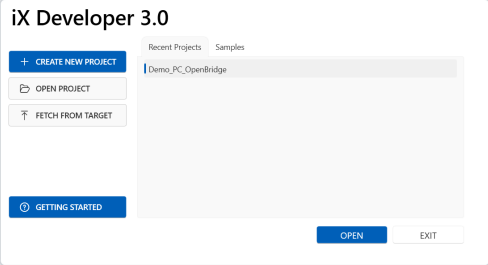
When you launch a newly installed version of iX Developer on a development PC with an older version of the software, you will be prompted to import settings from the previous version. The following settings can be imported:

- **License key**
- **Recent projects**
- **Added items in the Component Library Window**
- **Additional controls**

TIP
To import settings from a previous version at a later time, go to **File** > **Import settings**.

IMPORTANT
Starting from iX Developer 3.0, importing projects from **Information Designer** and **H-Designer/ADP** is no longer supported.

2.3. The Start Screen

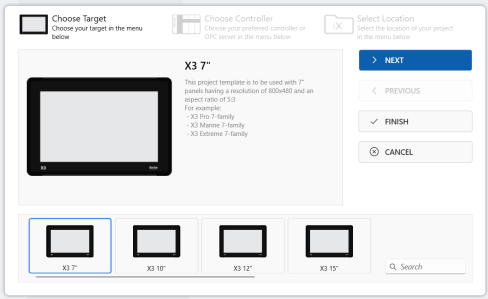


When you launch iX Developer, the **Start Screen** appears with the following options:

- **CREATE NEW PROJECT**: Create a new project using the **Wizard**. See Create a New Project With the Wizard.
- **OPEN PROJECT**: Open a recent project or sample project.
- **FETCH FROM TARGET**: Fetch a project from a target. See Fetch the Project From the HMI Panel.
- **GETTING STARTED**: Link to the Get Started chapter in this **User Guide**.

TIP
You can also create and open projects later using the File Menu.

2.4. The Wizard



The **Wizard** provides an easy way to create new projects. To open the **Wizard**:

1. Run iX Developer.
2. From The Start Screen, select **CREATE NEW PROJECT**.

2.4.1. Create a New Project With the Wizard

To create a new project with The Wizard, follow these steps:

1. Run iX Developer.
2. From The Start Screen, select **CREATE NEW PROJECT**.

3. Choose **Target** from the available options.

> **NOTE**
> **Target rotation** for HMI panels is not yet supported for X3 HMI panels.

4. ***For PC***: Select **Resolution** from the dropdown.

> **TIP**
> You can adjust the resolution later in the Project Properties.

5. Click **Next**.
6. Choose the **Brand** and **Protocol** for the Controller, or select OPC UA Server.

> **TIP**
> - To design and test a project directly on the development PC, without a controller connection, use the DEMO Controller. This controller includes standard tags and counters.
> - To change the controller later in the project, refer to the Add a Controller chapter.

> **NOTE**
> **OPC Classic** is no longer supported in iX Developer. Use OPC UA Server instead.

7. Click **Next**.
8. Enter the project **Name** and specify the **Location** for the project files.

> **NOTE**
> Do not save your iX project in a folder that syncs with **Microsoft OneDrive**, **Google Drive**, or similar services. This may prevent the project from opening and cause an **Access Denied** error.

9. Click **Finish**.

2.5. Integrate Customer-Specific Features

If supported, iX Developer allows integration of customer-specific features after installation.

Instructions

1. Go to **File** > **About**, click **Show Features** to display enabled features.

2. Enter the feature **Registration Key** under **Configured Features** and click **OK**.
3. Restart iX Developer to activate the new feature.

2.6. Update Software and Drivers

It is recommended to apply all available updates before starting to work on a project.

Update iX Developer:

- Open the File Menu and click **Update Software**.

Update OS3:

- Refer to the OS3 User Guide

Update controller drivers:

- Refer to the Driver Manual for the corresponding controller.

3. User Interface

This section describes the iX Developer user interface, which consists of the following parts:

File Menu

Description: Provides commands for managing projects, updating software, and configuring system settings.

Position: Top left of the screen

Quick Access Toolbar

Description: Contains frequently used controls for quick access.

Position: Top left of the screen.

Ribbon Tabs

Description: Organized into **Ribbon Groups**, each containing relevant **Controls** for managing screens and project functions.

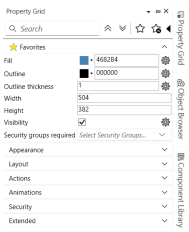
Position: Top of the screen.

Editor Area

Description: Used to create and manage key project components such as screens, tags, functions and scripts.

Position: Central workspace.

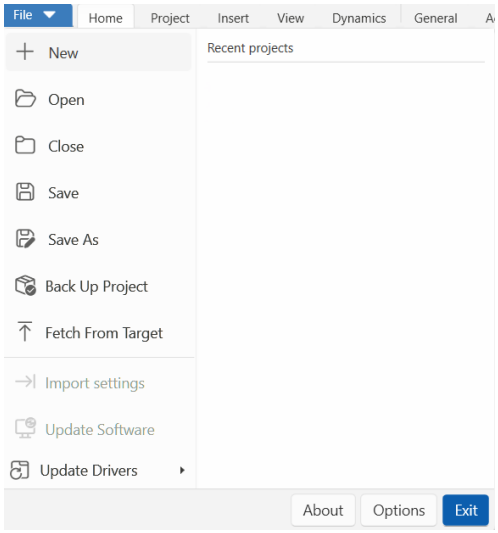
Tool Windows



Description: Includes various project management windows like the **Property Grid**, **Object Browser** and **Project Explorer**.

Position: Default docked around the Editor Area, but can be moved, see Tool Window Layout.

3.1. File Menu



The **File** menu in iX Developer provides commands for managing projects, updating software, and configuring system settings.

Parameter	Description
New	Opens the **Wizard**. See Create a New Project With the Wizard.
Open	Opens an existing project.
Close	Closes the current project.
Save	Saves the project.
Save As*	Saves the project with a different name and/or in a different location.
Back Up Project	Compresses the project into a ZIP file. See Back Up the Project.
Fetch from Target	Fetches a compressed project ZIP file from a connected target. See Fetch the Project From the HMI Panel.
Import Settings	Imports settings from a previous version of iX Developer.
Update Software	Installs the latest updates and patches. See Update Software and Drivers.
Update Drivers	Installs the latest drivers. See Update Software and Drivers.
About	Displays version details, registration key and enabled features.
Options	Opens the iX Developer Options.
Exit	Quits iX Developer.

*** Save As recommendations**

- To prevent changes to the original project, use **Save As** to create a new project file before making modifications.
- If managing many project files, it's recommended to create a new folder when using the **Save As** command.
- Using **Save As** does not prevent changes to the original project. To keep the original version unchanged, it is recommended to create a backup copy before making modifications.

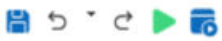
3.1.1. iX Developer Options

This **Options** window allows you to configure various application settings. To open it, go to **File** > **Options**.

Option		Description
Language options	**Language**	Selects the application language. Restart iX Developer for changes to take effect.
Build options	**CheckForUnusedTags**	Scans for unused tags and prompts for their removal. For further details, see Remove Unused Tags.
	ScriptWarningsBehaviour	Defines how script warnings are handled: • **Ignore** – No warnings displayed. • **Show** – Warnings are shown. • **TreatAsErrors** – Warnings are treated as errors.
Update Software options	**EnableCertificateVerification**	Verifies driver certificates during installation. Default: **True**. **IMPORTANT** Disabling this option may compromise system security but could be necessary if a driver cannot be installed due to a missing or invalid signature. To ensure maximum security, it is recommended to keep this option enabled.
	ShowUpdateInfoOnStartup	Displays update reminders on startup. Default: **True**
Script Debugger options	**Arguments** *(available only when **DebuggerType** is set to **Custom**)*	Defines the file insertion placeholder "%1" for custom debugger paths.
	Debugger	Specifies the path to the debugger application.
	DebuggerType	Selects the debugger type: • **Default** – Uses the built-in debugger. • **Custom** – Allows manual selection of a debugger path.
Transfer options	**TransferAllSourceCode**	Automatically enables **Include source code when transferring project** for new projects. See Project Transfer for details.
Message Info options	**ShowVersionInfoOnStartup**	Displays version update notifications when the application starts.

Option		Description
Ribbon options	**RibbonStrategy**	Defines how Ribbon Tabs behave when selecting objects in the workspace: • **GoToDefaultRibbonStrategy** - Changes the active ribbon tab based on the selected object. Selecting an object opens the General Ribbon Tab. • **SimpleRibbonStrategy** (default) - Keeps the current ribbon tab active. Selecting an object opens the Home Ribbon Tab. Double-clicking an object opens the General Ribbon Tab, except for specific objects like buttons, which open the Actions Ribbon Tab.
Screen Editor options	**EnableInPlaceEditing**	Allows direct text editing on objects in the workspace.
	ShowPanelAround	Displays an outline of the selected panel model in Layout View Mode.
	ShowSizeAndDistanceInformation	Shows object size and distance when resizing or moving objects.
	ShowToolTips	Displays tooltips.
Minitoolbar options	**ShowMiniToolbarAutomatic**	• **True**: Displays the Mini Toolbar automatically when selecting an object. • **False** (Default): Displays the Mini Toolbar only when accessed via the context menu.

3.2. Quick Access Toolbar

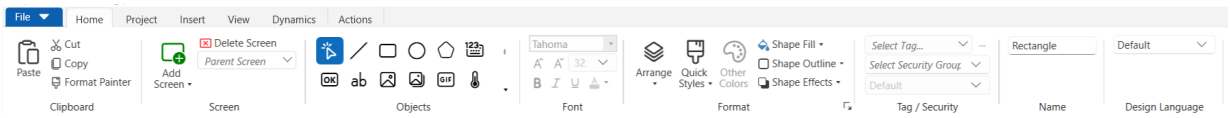


The **Quick Access Toolbar** provides quick access to frequently used commands for efficient project management.

Command	Description
Save	Saves the current project.
Undo	Reverses the last action. Click the arrow to choose from a list of recent actions.
Redo	Reverses the last **Undo** action.
Run	Run project. See Build, Rebuild, Run or Simulate.
Simulate	Simulate project. See Build, Rebuild, Run or Simulate.

TOOLTIPS
Hover your mouse over a control or command to view a brief description.

3.3. Ribbon Tabs



Each **Ribbon tab** contains one or more **Ribbon Groups**, which include **Controls** used for managing screens and project functions.

TOOLTIPS

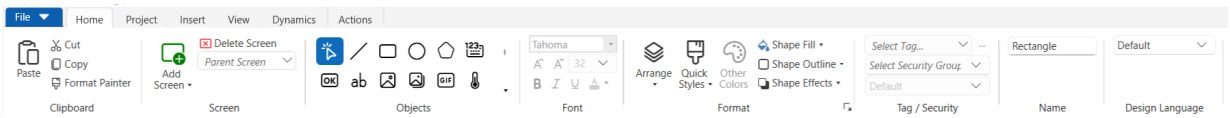
Hover your mouse over a control or command to view a brief description.

You can **minimize** or **expand** ribbon tabs in two ways:

- **Double-click** any ribbon tab.
- **Right-click** anywhere in the ribbon area, and choose **Minimize the Ribbon** from the context menu.

You can configure context-dependent behavior for ribbon tabs from the **Ribbon options** in iX Developer Options.

3.3.1. Home Ribbon Tab



The **Home** ribbon tab includes controls for editing screens within the project.

Ribbon groups in the Home ribbon tab

- Clipboard Group
- Screen Group
- Objects Group
- Font Group
- Format Group
- Tag/Security Group
- Name Group
- Design Language Group

3.3.1.1. Clipboard Group

The **Clipboard** group contains the following controls:

Control	Description	Shortcut
Paste	Copies the clipboard to the current screen.	**Ctrl** + **V**
Cut	Cuts the selection to the clipboard.	**Ctrl** + **X**
Copy	Copies the selection to the clipboard.	**Ctrl** + **C**

Control	Description	Shortcut
Format Painter	Copies formatting from one object to another. This is useful when creating a uniform appearance for objects. For more details, see Copy and Apply Formatting.	

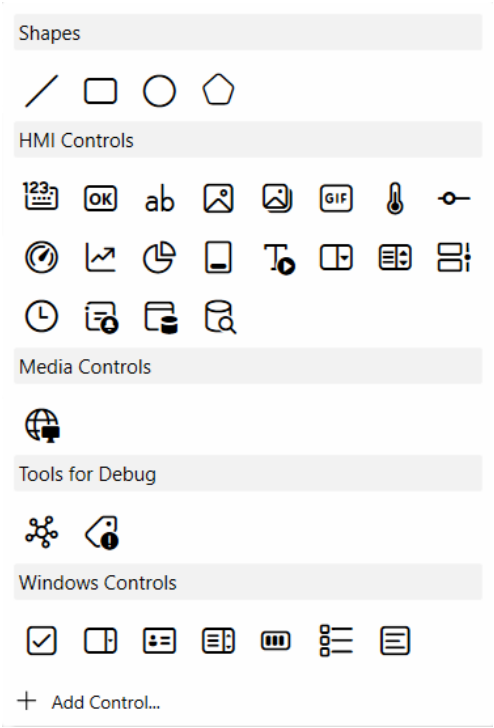
3.3.1.2. Screen Group

The **Screen** group contains the following controls:

Control	Description
Add Screen	Creates a new empty screen that opens for editing. For detailed information, see Add a Screen to the Project. Clicking the lower part of the **Add Screen** button opens a window with the available Manage Screen Templates
Delete Screen	Removes the current screen from the project and disconnects any associations to other screens. **NOTE** A deleted screen cannot be restored with the **Undo** command.
Background Screen	Any screen can be set as a background to the current screen. All objects in a background screen will function as expected in the project application. Objects in a background screen must be edited in their original screen. In the development environment, background screen objects will appear dimmed. For more information, see Set a Screen as Background or Foreground

TIP
For more information about screens, see the Screens section.

3.3.1.3. Objects Group



The **Objects** group contains static and dynamic objects that can be added to a screen.

> **TIP**
> For more information about objects, see the Objects section.

Object Variations Group

To display the **Object Variations** group, click an object in the Objects Group. If there are object variations available for that object , the **Object Variations** group appears to the right of the **Objects** group.

Many objects allow you to apply a preset style using the **Object Variations** feature. The Button Object and the Circular Meter Object offer a wide range of default styles, while most other objects support user-defined styles.

To add user-defined styles in the **Object Variations** menu, see chapter Create and Apply Object Variations.

3.3.1.4. Font Group

The **Font** group allows you to change the text style for the currently selected object. All fonts available in the development PC are accessible when designing the project. These fonts are included when the project is downloaded to the target device.

NOTE

- Keep in mind that the number of fonts in your project affects the total project size, and some fonts may require significant memory.
- The default font **Tahoma** does not support Chinese characters in the iX App. To display Chinese characters, you must switch to a compatible font, such as **MS Song**.
- If a Text Object has **Auto Stretch Font** enabled, the font size adjusts automatically when you resize the object's height. In this case, manual font settings do not apply.

3.3.1.5. Format Group

The **Format** group provides controls for adjusting the color and position of selected objects. Predefined quick styles allow for a uniform appearance across different objects. When multiple objects are selected, these controls can be applied to all selected objects simultaneously.

Clicking the small arrow in the lower right corner of the **Format** group opens additional format settings, such as **shadow**, **fill**, and **outline** properties.

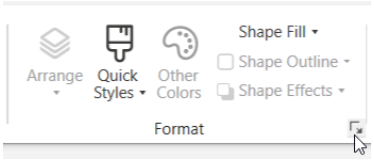
You can use the **Format Painter** control in the Clipboard Group to copy color and font properties from one object and apply them to another. For more information, see Copy and Apply Formatting.

Format group controls

Control	Description
Arrange	Contains commands for ordering, grouping, resizing, aligning, and distributing screen objects. For more information, see Arrange Objects.
Quick Styles	Provides preset visual object color schemes that can be applied to screen objects. These styles ensure a uniform appearance on items like meters and trend viewers. Quick styles are available for all iX Developer projects. An object with a style applied can be further customized, e.g., with font settings in the Font Group. **TIP** Controls in **General** > **Style** can reshape objects like meters, sliders, and trend viewers to predefined styles. These styles are not the same as the **Quick Styles**.
Other Colors	Allows customization of specific object properties, such as scale color for a Circular Meter Object or Linear Meter Object, or out-of-range validation for a Analog Numeric Object.
Shape Fill	Alters the fill color and gradient of the selected shape. A preview is shown directly on the object as you hover over different color options.
Shape Outline	Adjusts the outline color and width of the selected shape.

Control	Description
Shape Effects	Applies bevel styles and shadow effects. **NOTE** Bevel in combination with fill color **no fill** is only supported for iX TxC and PC targets. The bevel will not be visible on HMI panel targets.

Additional Format Properties



The Format Group includes additional properties that can be configured by clicking the arrow in the lower-right corner of the ribbon group. The available properties depend on the selected object.

3.3.1.6. Tag/Security Group

The **Tag/Security** group contains controls for managing controller tag connections, regulating object access based on security groups, and determining object visibility.

Controls	Description
Select Tag	Connects to a tag from the tag configuration page. Tags can also be added directly but will remain internal until linked to a controller tag. Click **...** to configure Expressions.
Select Security Groups	Limits access to the object based on security groups. If this field is left empty, the object is accessible to everyone.
Select Visibility	Defines object visibility: • **Default**: The visibility settings specified in **Dynamics** > **General** > **Visibility** is used. • **Disabled**: The object is disabled for all users. • **Hidden**: The object is invisible to users with insufficient access rights. • **Normal**: The object appears for all users.

TIP

- If access rights are not met, use the Security Function to configure an access denied message or to open a login dialog.
- For more information about access and security, see the Security section.

3.3.1.7. Name Group

When created, all objects and screens are automatically assigned names. These names can be viewed and edited in the **Name** group.

An object's name serves as a symbolic identifier and must follow specific conventions: it must be an alphanumeric string starting with a letter. Valid characters include letters (a-z, A-Z), numbers (0-9), and underscores ('_').

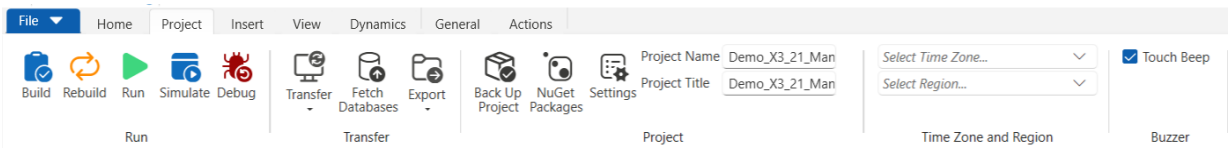
IMPORTANT
Renaming objects and screens that are referenced in Scripts may cause the scripts to malfunction.

3.3.1.8. Design Language Group

The **Design Language** group includes controls for managing the language settings of the project.

Control	Description
Current Language	Displays the language currently in use while designing the project. If the project does not support multiple languages, only the **Default** option is available.

3.3.2. Project Ribbon Tab



The **Project** ribbon tab includes controls for various project related functions.

Ribbon groups in the Project ribbon tab

- Run Group
- Transfer Group
- Project Group
- Time Zone and Region Group
- Buzzer Group

3.3.2.1. Run Group

The **Run** group contains the following commands that enables validation, rebuilding, simulation and debugging of the project. This table highlights the differences between these commands:

Command	Validates the script code	Compiles the project	Deletes temporary files	Runs the iX App on the development PC	Communicates with the controller(s)	See also
 Build	Yes	Yes Only changes are recompiled.	No	No	No	Build, Rebuild, Run or Simulate
 Rebuild	Yes	Yes Entire project is recompiled.	Yes	No	No	Build, Rebuild, Run or Simulate
 Run	Yes	Yes	Yes	Yes	Yes	Build, Rebuild, Run or Simulate
 Simulate	Yes	Yes	Yes	Yes	No	Build, Rebuild, Run or Simulate
 Debug	Yes	No	No	No	No	Debug Script

3.3.2.2. Transfer Group

The **Transfer** group contains controls that transfers the project to an HMI panel, fetches the database from an HMI panel and exports the project to a folder.

Control	Description
Transfer	Transfers the project as an app to an HMI panel within the network environment. For more information, see Transfer the iX App to the HMI Panel.
Fetch Databases	Collects the database from an HMI panel in the network environment to a file on the development PC. For more information, see Fetch Databases From the HMI Panel.
Export	Exports the project to a user-selected folder. For more information, see Export the iX App.

3.3.2.3. Project Group

The **Project** group allows editing of the project target, name, and title. The project title is displayed in the title bar in the iX App by default. If the title is left empty, the screen name/screen title will be displayed.

Control	Description
Back Up Project	Compresses the project source file into a ZIP file. The ZIP file can be saved to a USB stick or directly to a folder on a runtime PC if a network connection is configured. For more information, see Back Up the Project.
NuGet Packages	iX Developer 3.x contains the feature to add and use third-party **NuGet packages**, replacing the old **Referenced Assemblies** feature from iX Developer 2.x. For more information on NuGet, refer to the NuGet Packages section.

Control	Description
Settings	Opens the Project Properties.
Project Name	Displays the project name, allowing you to change it. **NOTE** When renaming a project, you can not undo the action.
Project Title	Displays the project title, allowing you to change it.

3.3.2.4. Time Zone and Region Group

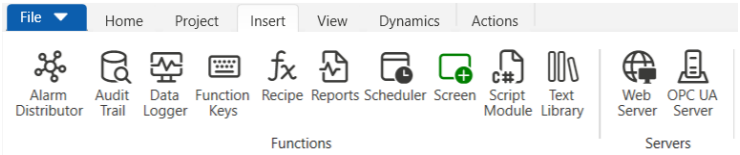
The **Time Zone and Region** group allows you to configure the time zone and region settings for your project.

Control	Description
Select Time Zone	Sets the time zone for the project.
Select Region	Sets the region settings for the project.

3.3.2.5. Buzzer Group

The **Buzzer Group** contains the **Touch Beep** control. Enable this option to activate a key beep sound during keyboard input in the iX App.

3.3.3. Insert Ribbon Tab



The **Insert** ribbon tab contains controls for adding or removing functions in the Functions Section and servers in the Servers Section.

Ribbon groups in the Insert ribbon tab:

- Functions Group
- Servers Group

3.3.3.1. Functions Group

The **Functions** group contains functions that can be added to the Functions Section in the Project Explorer Window. Click on the symbol to add the desired function.

Available functions

- Alarm Distributor Function
- Alarm Server Function
- Audit Trail Function
- Data Logger Function
- Function Keys Function

- Multiple Languages Function
- Recipe Function
- Reports Function
- Scheduler Function
- Security Function
- Tags Function
- Text Library Function

TIP
For detailed information about functions, refer to the Functions section.

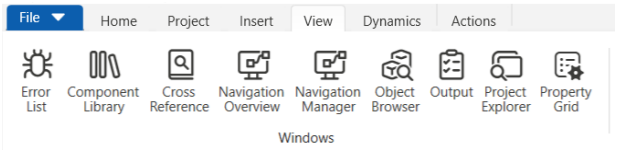
3.3.3.2. Servers Group

The **Servers** group contains the following controls:

Control	Description
Web Server	Enables hosting of project files for a web client.
OPC UA Server	Allows publishing of tags with an OPC UA Server.

TIP
For more information about servers, refer to the Servers section.

3.3.4. View Ribbon Tab



The **View** ribbon tab contains controls for toggling the visibility of the Tool Windows.

Ribbon groups in the View ribbon tab:

- Windows Group

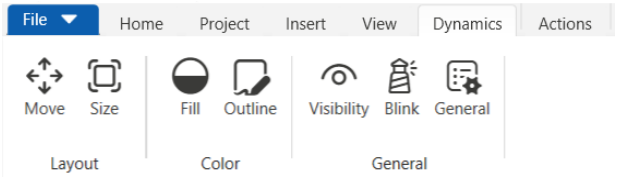
3.3.4.1. Windows Group

The **Windows** group contains controls that show and hide the Tool Windows.

Window	Description	Link
Error List	Displays information about problems and errors detected during build.	Error List Window
Component Library	Contains predefined as well as user-defined graphical components.	Component Library Window

Window	Description	Link
Cross Reference	Provides an overview of where a specific tag is used.	Cross Reference Window
Navigation Overview	Provides an overview of all screens included in the project, with zooming facilities.	Navigation Overview Window
Navigation Manager	Manages screens and screen navigation.	Navigation Manager Window
Object Browser	Displays an overview of all objects in the current screen and enables setting visibility and locking objects easily, as well as placing objects in front or back of each other.	Object Browser Window
Project Explorer	Shows all screens and components included in the application.	Project Explorer Window
Output	Displays detailed information concerning the project validation and build.	Output Window
Property Grid	Shows detailed properties of a selected screen or object.	Property Grid Window

3.3.5. Dynamics Ribbon Tab



The **Dynamics** ribbon tab includes controls that are used for changing the properties of an object depending on a changed tag value.

Ribbon groups in the Dynamics ribbon tab:

- Layout Group
- Color Group
- General Group

3.3.5.1. Layout Group

The **Layout** group contains the following controls:

Control	Description
Move	Changes an object's position, based on a tag value. For more information, see Move Object Based on Tag Value Changes.
Size	Changes an object's size, based on a tag value. For more information, see Resize Object Based on Tag Value Changes.

3.3.5.2. Color Group

The **Color** group contains the following controls:

Control	Description
Fill	Change an object's fill color based on a tag value. For more information, see Change Object Fill Color Based on Tag Value.
Outline	Change an object's outline (border) color based on a tag value.

3.3.5.3. General Group

The **General** group contains the following controls:

Control	Description
Visibility	Determines if an object is to be displayed in the screen or not, based on a tag value. For more information, see Hide Object when Tag Value Changes.
Blink	Enables or disables blinking, based on the tag value. For more information, see Blinking Objects.
General	Click to open the Edit General Dynamics Dialog.

Edit General Dynamics Dialog

The **Edit General Dynamics** dialog contains different properties depending on the selected object. To open the **Edit General Dynamics** dialog, click **Dynamics** > **General** > **General**.

Parameter	Converter	Object	Description
Acknowledge all	·bool	Alarm Viewer Object	Set visibility for the **Acknowledge All** button.
Acknowledge Visible	·bool	Alarm Viewer Object	Set visibility for the **Acknowledge Visible** button.
Acknowledge Selected	·bool	Alarm Viewer Object	Set visibility for the **Acknowledge Selected** button.
Active	·bool	Chart Object	Activate or stop the data input.
Address	·raw ·text library group	Web Browser Object	Change the default URL.
Alternate background color	·color interval	Animated Label Object	Set background color.
Alternate foreground color	·color interval	Animated Label Object	Set foreground color.
Animation speed	·linear	Animated Label Object	Set animation speed.
Auto-scale X-axis	·bool	Chart Object	Toggle automatic calculation of the minimum and maximum X-axis values.
Auto-scale Y1-axis	·bool	Chart Object	Toggle automatic calculation of the minimum and maximum Y1-axis values.
Auto-scale Y2-axis	·bool	Chart Object	Toggle automatic calculation of the minimum and maximum Y2-axis values.

Parameter	Converter	Object	Description
Auto-size	·bool		Resize the object according to the size of the default text string length and font size.
Button color	·color interval	Alarm Viewer Object	Set the button color.
Button height	·raw ·linear	Alarm Viewer Object	Set the height of the buttons.
Button width	·raw ·linear	Alarm Viewer Object	Alter the width of the buttons.
Clear	·bool	Alarm Viewer Object	Set visibility for the **Clear** button.
Clear Visible	·bool	Alarm Viewer Object	Set visibility for the **Clear Visible** button.
Clockwise	·bool	Circular Meter Object	Set the rotation orientation.
Column header color	·color interval		Set the column header background color.
Enabled	·bool		Enable or disable an object.
End angle	·raw ·linear	Circular Meter Object	Set the end angle.
Filter	·bool	Alarm Viewer Object	Set visibility for the filter button.
Font color	·color interval		Set font color in for instance the text object.
Font size	·raw ·linear		Set font size in for instance the text object.
Group background	·color interval	Action Menu Object	Set the group background color.
Group foreground	·color interval	Action Menu Object	Set the group foreground color.
Header	·raw ·text library group	Chart Object	Set the header.
High Region Color	·color interval	Circular Meter Object	Set the high region color.
High Region Max	·raw ·linear	Circular Meter Object	Set the maximum value of the high region indicator.
High Region Min	·raw ·linear	Circular Meter Object	Set the minimum value of the high region indicator.
Indicator color	·color interval		Set the indicator color in for instance the circular meter object.
Info	·bool	Alarm Viewer Object	Set visibility for the info button.

Parameter	Converter	Object	Description
Lock aspect ratio	·bool	Multi Picture Object	Used with the multi picture object to toggle the height/width ratio of the object.
Low region color	·color interval	Circular Meter Object	Set the low region color.
Low region max	·raw ·linear	Circular Meter Object	Set the maximum value of the low region indicator.
Low region min	·raw ·linear	Circular Meter Object	Set the minimum value of the low region indicator.
Lower limit	·raw ·linear	Analog Numeric Object	Set the minimum value that can be entered.
Major ticks	·raw ·linear		Set the number of labeled time marks in for instance the linear meter object.
Maximum	·raw ·linear		Set the end value of the scale in for instance the linear meter object.
Major value tics	·raw ·linear	Trend Viewer Object	Set the number of labeled scale marks on the scale.
Max value	·raw ·linear	Trend Viewer Object	Set the end value of the scale.
Min value	·raw ·linear	Trend Viewer Object	Set the start value of the scale.
Minor value ticks	·raw ·linear	Trend Viewer Object	Set the number of scale ticks between two adjacent major ticks on the scale.
Medium region color	·color interval	Circular Meter Object	Set the medium region color.
Medium region max	·raw ·linear	Circular Meter Object	Set the maximum value of the medium region indicator.
Medium region min	·raw ·linear	Circular Meter Object	Set the minimum value of the medium region indicator.
Minimum	·raw ·linear		Set the start value of the scale in for instance the linear meter object.
Minor ticks	·raw ·linear		Set the number of scale ticks between two adjacent major ticks.
Opacity	·raw ·linear		Set the opacity of for instance the PDF viewer object.
Outline thickness	·raw ·linear		Set the outline thickness of for instance the rectangle object.
Picture height	·raw ·linear	Button Object	Set picture height.

Parameter	Converter	Object	Description
Picture name	·raw ·text library group	Picture Object	Set the picture shown in the picture object.
Picture stretch	·bool	Button Object	Stretch a picture.
Picture width	·raw ·linear	Button Object	Set the picture width.
Play/Pause	·bool	Alarm Viewer Object	Set visibility for the pause button.
Prefix	·raw ·text library group	Analog Numeric Object	Set a prefix to the digits entered.
Radius X	·raw ·linear	Rectangle Object	Set the radius of the X axis in the rectangle object.
Radius Y	·raw ·linear	Rectangle Object	Set the radius of the Y axis.
Scale color	·color interval	Circular Meter Object	Set the scale color.
Scale	·bool	Circular Meter Object	Set visibility for the scale.
Selected background	·color interval	Action Menu Object	Set the background color of the selected action.
Selected foreground	·color interval	Action Menu Object	Set the foreground color of the selected action.
Selected group background	·color interval	Action Menu Object	Set the background color of the selected menu group.
Selected group foreground	·color interval	Action Menu Object	Set the foreground color of the selected menu group.
Separator color	·color interval		Set the separator color.
Show header	·bool	Chart Object	Set visibility for the header.
Show legend	·bool	Chart Object	Set visibility for the legend.
Read only	·bool		Control whether for instance an analog numeric object editable or not.
Show X-axis grid	·bool	Chart Object	Set visibility for the X Axis grid.
Show X-axis	·bool	Chart Object	Set visibility for the X Axis.
Show Y1-axis grid	·bool	Chart Object	Set visibility for the Y1 Axis grid.
Show Y1-axis	·bool	Chart Object	Set visibility for the Y1 Axis.
Show Y2-axis grid	·bool	Chart Object	Set visibility for the Y2 Axis grid.
Show Y2-axis	·bool	Chart Object	Set visibility for the Y2 Axis.

Parameter	Converter	Object	Description
Start angle	·raw ·linear	Circular Meter Object	Set the start angle.
Stretch	·bool		Adjust the height and width to the selection frame in for instance the picture object.
Suffix	·raw ·text library group	Analog Numeric Object	Set a suffix to the digits entered in the analog numeric object.
Text	·raw ·text library group		Set the text shown in for instance the text object.
Title	·raw ·text library group	Roller Panel Object	Set the title.
Title background color	·color interval	Roller Panel Object	Set the title background color.
Title foreground color	·color interval	Roller Panel Object	Set the title foreground color.
Transparent	·bool		Set the transparency of for instance the button object.
Upper limit	·raw ·linear	Analog Numeric Object	Set the maximum value that can be entered.
Value	·raw ·linear	Progress Bar Object	Set the value.
X1	·raw ·linear	Linear Meter Object	Set the X1 pixel coordinates
X2	·raw ·linear	Linear Meter Object	Set the X2 pixel coordinates
X3	·raw ·linear	Linear Meter Object	Set the X3 pixel coordinates
X4	·raw ·linear	Linear Meter Object	Set the X4 pixel coordinates
X-axis grid color	·color interval	Chart Object	Set the grid color of the X axis.
X-axis max value	·raw ·linear	Chart Object	Set the maximum value of the X axis.
X-axis min value	·raw ·linear	Chart Object	Set the minimum value of the X axis.
Y1-axis grid color	·color interval	Chart Object	Set the grid color of the Y1 axis.
Y1-axis max value	·raw ·linear	Chart Object	Set the maximum value of the Y1 axis.

Parameter	Converter	Object	Description
Y1-axis min value	·raw ·linear	Chart Object	Set the minimum value of the Y1 axis.
Y2-axis grid color	·color interval	Chart Object	Set the grid color of the Y2 axis.
Y2-axis max value	·raw ·linear	Chart Object	Set the maximum value of the Y2 axis.
Y2-axis min value	·raw ·linear	Chart Object	Set the minimum value of the Y2 axis.

Converter Operators

Converter operators modify object properties and can be selected from the Edit General Dynamics Dialog dialog.

Converter operator	Description
Bool converter	Converts 0 to false and all other values to true.
Color converter	Converts value intervals to specified colors.
Linear converter	Converts values according to the linear equation.
Raw converter	Converts the value to a new value that matches the target type.
Text Library Group converter	Converts a value to a text.

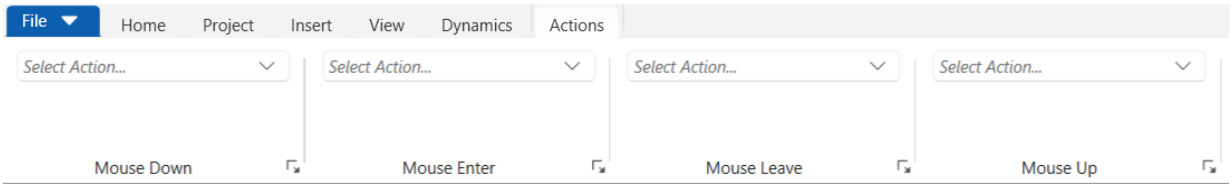
3.3.6. General Ribbon Tab

The **General** ribbon tab offers context-specific controls based on the selected Object. To access it, select an object, and the corresponding controls for the **General** ribbon tab will appear.

Examples of using the controls in the General ribbon tab:

- Modify the appearance of graphical shapes.
- Adjust scale settings for meters and trend viewers.
- Configure additional settings for specific objects.

3.3.7. Actions Ribbon Tab

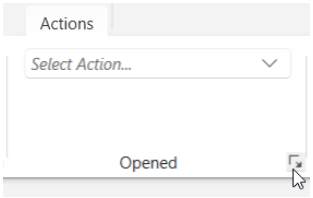


The **Actions** ribbon tab ribbon tab provides context-specific controls to define actions based on the selected Object. To access it, select an object, and the corresponding controls for the **Actions** ribbon tab will appear.

An **action** is a task performed by the system, such as closing a screen. An **action trigger** is an event that generates an action, for example, clicking a button.

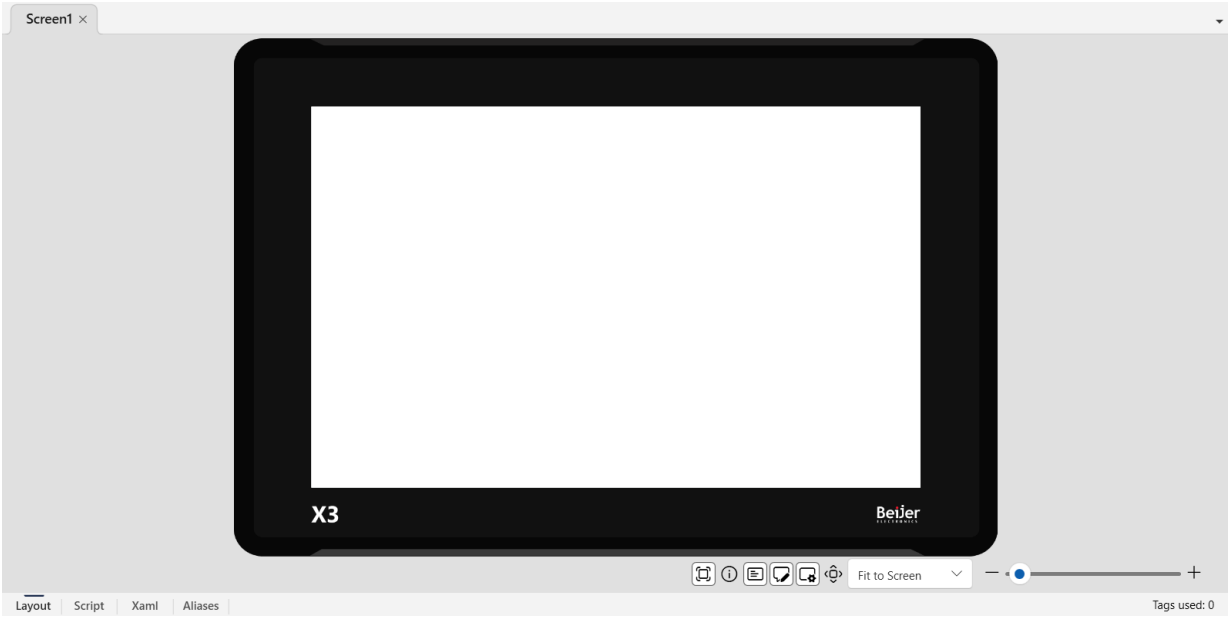
TIP
For detailed information about actions, see the Actions section.

3.3.7.1. Additional Actions Properties



The ribbon groups in the Actions Ribbon Tab includes additional properties that can be configured by clicking the arrow in the lower-right corner of the ribbon group. The available properties depend on the selected object.

3.4. Editor Area



The **Editor Area** is the central workspace of iX Developer, it is used to create and manage key project components such as screens, tags, functions and scripts.

TIP
For information on how to zoom and pan in the editor area, see Zoom and Pan.

3.4.1. Editor View Modes

The **Editor View Modes** are located in the the lower left part of the Editor Area.

- Layout

- Script
- Xaml
- Aliases

3.4.1.1. Layout View Mode

The **Layout** view mode, the main editing window, is where screens are edited.

TIP
For a list of keyboard shortcuts, see Keyboard Shortcuts.

Layout View Mode Controls

The **Layout View Mode Controls** are located in the lower-right corner of the Editor Area and manage object appearance, screen size, and screen layout.

Control	Description
Show/Hide Size and Distance Information	Displays the object's size during resizing and shows the distance to desktop edges when moving the object.
Show/Hide Info	Shows tag information for each object and indicates if dynamics or actions are configured.
Show/Hide Screen Tooltips	Displays semi-transparent tooltips for objects when hovering over them. Does not affect menu command tooltips.
Enable/Disable In-place Editing	Allows editing an object directly without accessing its properties.
Show/Hide Target Frame	Displays a virtual HMI panel frame around the screen.
Fit to Screen	Maximizes the screen to fit the desktop area.
Zoom dropdown	Sets the zoom level by selecting a value from the list or choosing **Fit to Screen**.
Zoom slider	Adjusts the zoom by moving the slider or by clicking the **–** and **+** buttons.

3.4.1.2. Script View Mode

The **Script** view mode allows you to configure scripted actions for screens, objects, tags, function keys, and more. Scripts are written in **C#** syntax, offering a flexible method to implement custom functionality.

TIP
For information on scripting in iX Developer, refer to the Scripts section.

3.4.1.3. Xaml View Mode

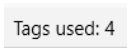
In the **Xaml** view mode, you can add and modify the XAML code. The **.xaml** files are stored in the Project Folder.

3.4.1.4. Aliases View Mode

In the **Aliases** view mode, you can configure aliases and instances.

TIP
For detailed information on aliases and instances, see Aliases.

3.4.2. Project Tag Usage

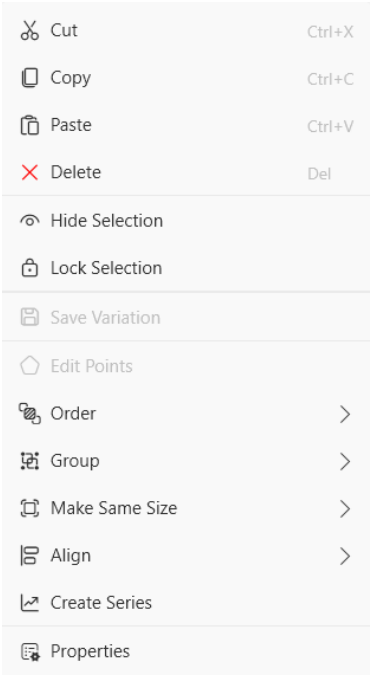


The total number of tags used in the project is displayed in the lower-right corner of the Editor Area.

3.4.3. Context Menu and Mini Toolbar

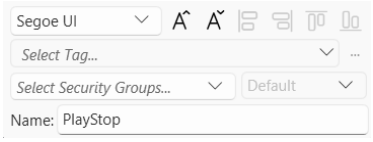
When you **right-click** Objects in the Editor Area, a context menu and a mini toolbar appear:

3.4.3.1. Context Menu



The **Context menu** contains standard Microsoft application commands, like **Copy**, **Paste**, and other basic commands.

3.4.3.2. Mini Toolbar



The **Mini toolbar** is a context-sensitive menu for Objects that includes commonly used controls from the Home Ribbon Tab. Its display behavior is controlled through the setting **ShowMiniToolbarAutomatic** in iX Developer Options.

3.5. Tool Windows

The **tool windows** are windows that can be opened in or around the Editor Area. You can open or close these windows from the View Ribbon Tab. To change the layout of the workspace by customizing the tool windows, see Tool Window Layout.

Available tool windows

Error List Window	Displays information about problems and errors detected during build. **Default position**: Docks to the lower part of the application window.
Component Library Window	Contains predefined as well as user-defined graphical components. **Default position**: Docks to the right part of the application window as a tab.

Cross Reference Window	Provides an overview of where a specific tag is used. **Default position**: Docks to the right part of the application window.
Navigation Overview Window	Provides an overview of all screens included in the project, with zooming facilities. **Default position**: Floating window.
Navigation Manager Window	Manages screens and screen navigation. **Default position**: Editor Area.
Object Browser Window	Displays an overview of all objects in the current screen and enables setting visibility and locking objects easily, as well as placing objects in front or back of each other. **Default position**: Floating window.
Project Explorer Window	Shows all screens and components included in the application. **Default position**: Docks to the left part of the application window.
Output Window	Displays detailed information concerning the project validation and build. **Default position**: Docks to the lower part of the application window.
Property Grid Window	Shows detailed properties of a selected screen or object. **Default position**: Docks to the right part of the application window as a tab.

3.5.1. Component Library Window

The **Component Library** contains components categorized into groups. These components include predefined or user-defined graphical objects and other files. To show the **Component Library**, click **View** > **Windows** > **Component Library**.

Some of the component types that may be included are:

- **Predefined components**
- **Objects**
- **Text objects** configured with multiple texts
- **Symbol objects** configured with multiple symbols
- **Grouped objects**
- **External pictures**
- **Text**
- **Folders** (file folders)
- **HTM/HTML files**
- **Media files**
- **PDF files**

3.5.1.1. Project Pictures

The **Project Pictures** section of the Component Library Window is project specific, and contains components that are used in the current project. The components in **Project Pictures** are saved in the Symbols Folder.

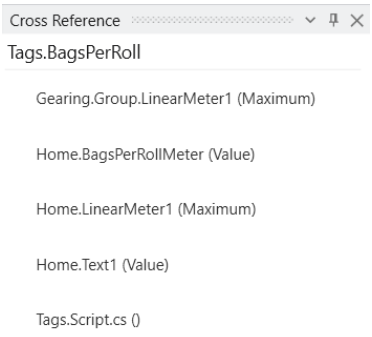
3.5.1.2. Project Files

The **Project Files** section in the Component Library Window is project-specific and contains files that have been added during the project creation. The **Project Files** folder is located in the Project Folder.

Project files and folders can be added or deleted from the **Project Files** folder before downloading the project to an HMI panel.

> **NOTE**
> Avoid using special characters (for example %, & or #) in filenames.

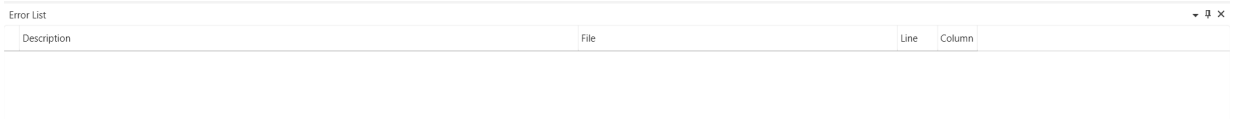
3.5.2. Cross Reference Window



The **Cross Reference** window provides an overview of where a specific tag is used in the current project. All occurrences of the tag within the current project are listed. **Double-clicking** an occurrence in the list automatically takes you to the object or function where the tag is used.

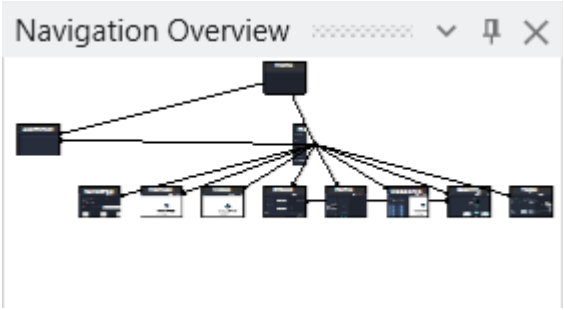
To display the **Cross Reference** window, click **View** > **Windows** > **Cross reference**. It can also be accessed via **Project Explorer** > **Functions** > **Tags** > **Cross Reference**, or by **right-clicking** a tag.

3.5.3. Error List Window



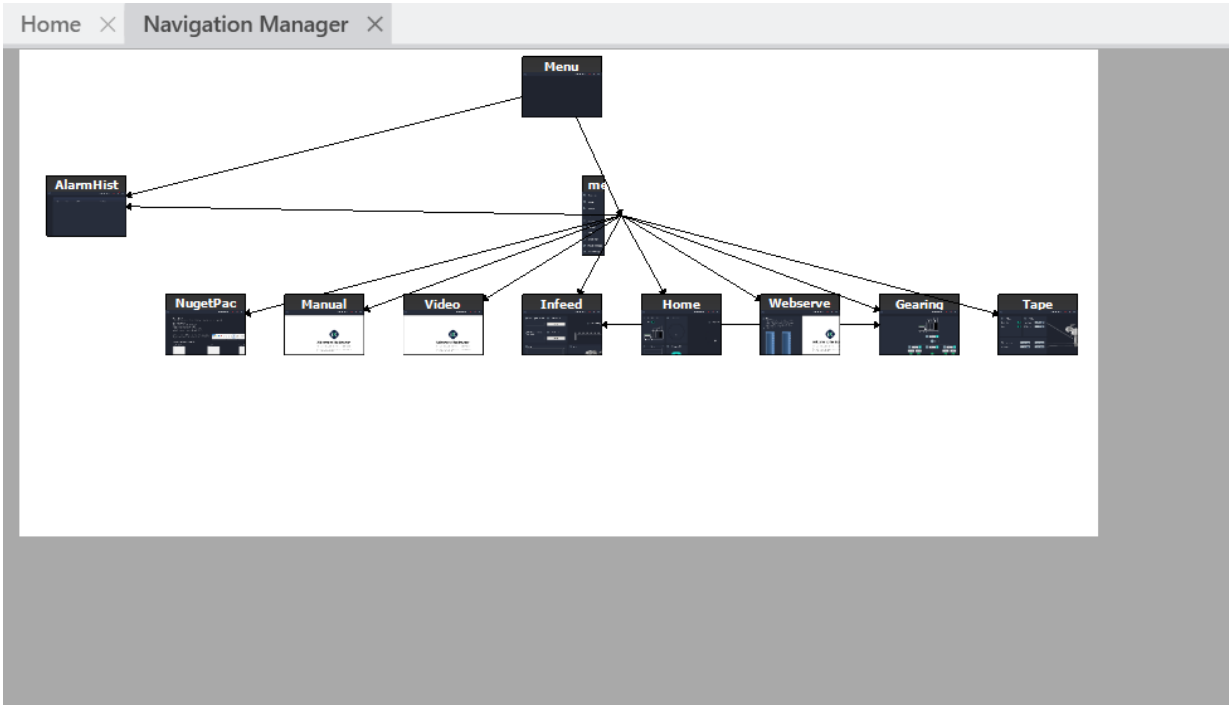
The **Error List** displays information about problems and errors detected in the current project. Build errors are reported in the Output Window. To view the **Error List**, click **View** > **Windows** > **Error List**.

3.5.4. Navigation Overview Window



The **Navigation Overview** window provides an overview of all screens included in the project. The **red square** controls the zoom level of the Navigation Manager Window. To display the **Navigation Overview** window, click **View** > **Windows** > **Navigation Overview**.

3.5.5. Navigation Manager Window



The **Navigation Manager** provides an overview of all screens in the current project and is used to manage screens and their relationships. It opens within the Editor Area. To display the **Navigation Manager** window, click **View** > **Windows** > **Navigation Manager**.

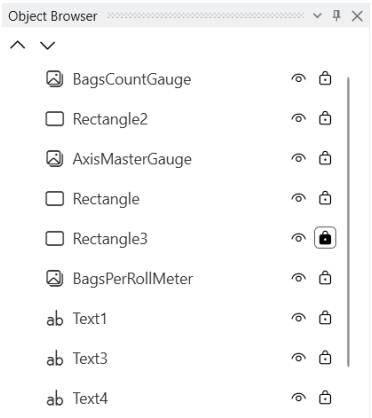
TIP
For detailed information on how to work with screens, see the Screens section.

3.5.5.1. Navigation Manager Context Menu

The context menu contains commands for editing in the Navigation Manager Window. To open the context menu, **right-click** anywhere in the **Navigation Manager**.

Parameter			Description
Show *(only when a screen is selected)*			Displays the selected screen.
Delete *(only when a screen is selected)*			Deletes the selected screen.
Link*	**Show Same Level References**		Filters out links to screens on the same hierarchical level.
	Show Back References		Shows navigation links to higher hierarchical levels. When not selected, only links to screens on lower levels, and links between screens on the same level are shown.
	Link type	**Directed**	Draws straight link lines between screens, using the shortest path.
		Orthogonal	Draws link lines in an orthogonal pattern.
* Changing the **Link** settings only affects the appearance in the **Navigation Manager**.			

3.5.6. Object Browser Window



The **Object Browser** window lists all objects on a screen and allows you to manage their visibility. To open the **Object Browser**, go to **View** > **Windows** > **Object Browser**.

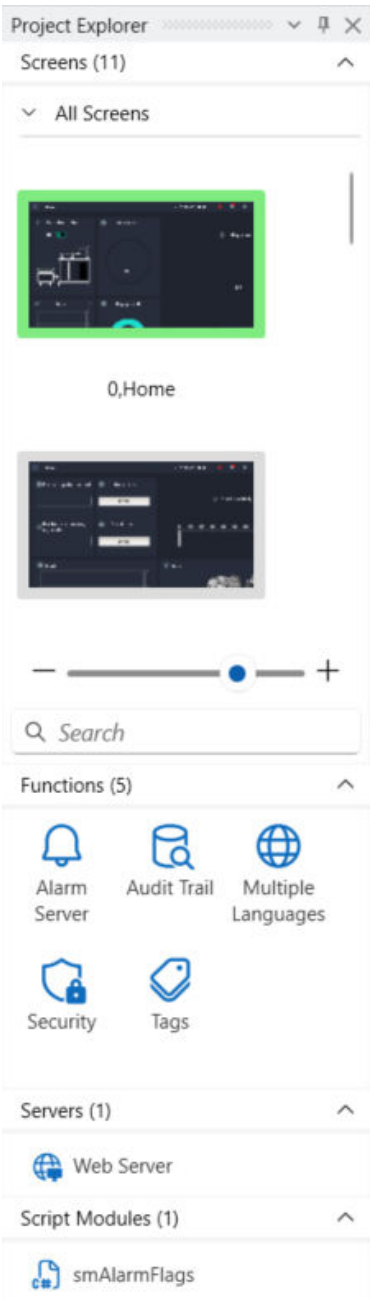
The order of objects in the **Object Browser** matches their front-to-back order on the screen, running from top to bottom.

- Use the **up arrow** to move objects up and bring them forward on the screen.
- Use the **down arrow** to move objects down and send them backward.
- The **eye** icon toggles the object's visibility.
- The **lock** icon toggles whether the object is locked.

3.5.7. Output Window

The **Output** window displays messages related to project validation and building. It lists created project files along with their folder paths and any errors encountered. To display the **Output** window, click **View** > **Windows** > **Output**.

3.5.8. Project Explorer Window



The **Project Explorer** window, located on the left side of the screen, is divided into several sections. To display the **Project Explorer**, click **View** > **Windows** > **Project Explorer**.

Usage instructions

- Click a section heading to expand or collapse the section.
- Right-click a section heading to access a subset of commands from the File Menu and the Insert Ribbon Tab.
- Right-click a component within a section to open a context menu with common operations.

> **NOTE**
> Not all commands are available for every component.

Project explorer sections

- Screens Section
- Functions Section
- Data Loggers Section
- Recipes Section
- Script Modules Section

3.5.8.1. Screens Section

The **Screens** section lists the screens included in the project. The startup screen is marked with a **green frame**. The number in parenthesis in the heading indicates the number of screens included in the project.

- **Click** on a screen to open it for editing in the Editor Area.
- **Right-click** on a screen to open a context menu with various operations.
- Use the **zoom slider** to zoom in and out.
- Use the **Search** area to search for screens.

> **TIP**
> For detailed information on how to work with screens, see the Screens section.

3.5.8.2. Functions Section

The **Functions** section contains predefined and user-defined functions.

Click on the function to open its The Configuration Pages.

Functions can be added or removed from the **Functions** section from the Insert Ribbon Tab. **Alarm Server**, **Multiple Languages**, **Security** and **Tags** are predefined and cannot be deleted.

Available functions

Function	Description	Link
Alarm Server	Handles alarm groups and alarm items.	Alarm Server Function
Alarm Distributor	Distributes alarms via e-mail between HMI panels.	Alarm Distributor Function

Function	Description	Link
Audit Trail	Enables tracking of operator actions.	Audit Trail Function
Function Keys	Sets function keys for HMI panels and PC keyboard.	Function Keys Function
Multiple Languages	Includes configuration for multiple languages for user texts and system texts. Texts can be exported, edited and reimported to the project. A tool for automatic translations is also included.	Multiple Languages Function
Reports	Allows adding *Excel* report templates to the project.	Reports Function
Scheduler	Controls events in the process at special times.	Scheduler Function
Screen	Click to add a new screen to the project.	Add a Screen to the Project
Security	Security management for users and groups of users.	Security Function
Tags	Handles data values referred to as **tags**.	Tags
Text Library	With the text library function, text tables can be created, where values are linked to texts.	Text Library Function

3.5.8.3. Script Modules Section

The **Script Modules** section allows you to program features not available through standard functions and actions or to share functionality across multiple functions.

To add a script module, click **Insert** > **Functions** > **Script Module**, or **right-click** anywhere in the script module section and select **Add Script Module**. You can also use the Apply Script Action to apply script modules to an object, just like other trigger actions.

TIP

For detailed information on scripting in iX Developer, refer to the Scripts chapter.

3.5.8.4. Data Loggers Section

The **Data Loggers** section is used to save and log data to a database at set intervals or when values change.

To add a data logger, click **Insert** > **Functions** > **Data Logger**, or **right-click** anywhere in the data loggers section and choose **Add Data Logger**.

TIP

For detailed information about data loggers, see the Data Logger Function chapter.

3.5.8.5. Recipes Section

The **Recipes** section handles a predefined set of recipe items. Several recipes can be inserted in the project.

To add recipes to the recipes section, click **Insert** > **Functions** > **Recipe**, or **right-click** anywhere in the recipes section and choose **Add Recipe**.

TIP
For detailed information about recipes, see the Recipe Function chapter.

3.5.9. Property Grid Window

The **Property Grid** window provides a detailed overview of an object's properties, offering more options than the ribbon tabs, such as object size, position, and input delay for mouse and touch screen in the iX App.

To display the **Property Grid** window, click **View** > **Windows** > **Property Grid**, or **right-click** an object and select **Properties**.

3.5.9.1. Symbols in the Property Grid Window

The property grid window has a **top menu bar**. To expand or minimize the top menu bar, click the black arrow on the right side of the menu.

Symbol	Description
	Show only the properties marked as favorites.
	Show all properties.
	Show all properties that do not have a default value.
	Show/hide the **Favorites** stars.
	Show/hide the Copy Properties Section and checkboxes.

Favorites

Frequently used properties can be added to the **Favorites** category for quick access. Favorites are saved per object type (e.g., the Button Object has its own set, and the Analog Numeric Object has another). To add a property to **Favorites**, click the star next to the property name. If the stars are not visible, click in the top menu.

Symbol	Description
	Property is added as a favorite.
	Property is not added as a favorite.

Copy Properties Section

The **Copy Properties** section of the Property Grid Window is used for copying properties from one object to one or more targets. To display the **Copy Properties** section, click  in the top menu bar.

Symbol	Description
	Select all properties for the object.
	Deselect all properties for the object.
	Copy the selected properties to the clipboard.

How to Copy Properties

Use the checkboxes next to the property name to select which properties to copy. Paste the values onto the target object(s) by selecting the target(s) and click **Home** > **Clipboard** > **Paste**, or by pressing **Ctrl** + **V**.

Only properties shared by both the source and target objects are copied. Actions, can only be copied between objects of the same type (e.g., from button to button).

4. Navigation

This chapter describes keyboard shortcuts, and zooming and panning methods for efficient navigation. It also explains how to manage tool windows for an optimized workspace. It covers docking, undocking, tabbed tool windows, and the Auto-Hide feature.

4.1. Keyboard Shortcuts

The following keyboard shortcuts are available in iX Developer:

File menu and Ribbon tabs

Shortcut	Description
Alt + **F**	Open the File Menu.
Alt + **H**	Go to the Home Ribbon Tab.
Alt + **P**	Go to the Project Ribbon Tab.
Alt + **I**	Go to the Insert Ribbon Tab.
Alt + **V**	Go to the View Ribbon Tab.
Alt + **D**	Go to the Dynamics Ribbon Tab.
Alt + **G**	Go to the General Ribbon Tab.
Alt + **A**	Go to the Actions Ribbon Tab.

View modes

Shortcut	Description
Ctrl + **P** + **1**	Go to the Layout View Mode.
Ctrl + **P** + **2**	Go to the Script View Mode.
Ctrl + **P** + **3**	Go to the Xaml View Mode.
Ctrl + **P** + **4**	Go to the Aliases View Mode.

Project execution

Shortcut	Description
F1	Open this manual. Press **F1** in Script View Mode for scripting help.
Alt + **4** or **F5**	Run project. See Build, Rebuild, Run or Simulate.
F6	Build project. See Build, Rebuild, Run or Simulate.
Alt + **5**	Simulate project. See Build, Rebuild, Run or Simulate.

Universal commands

Shortcut	Description
Alt + **2** or **Ctrl** + **Z**	Undo
Alt + **3** or **Ctrl** + **Y**	Redo
Ctrl + **A**	Select all
Ctrl + **X**	Cut
Ctrl + **C**	Copy

Shortcut	Description
Ctrl + **V**	Paste
Ctrl + **N**	New
Ctrl + **O**	Open
Alt + **1** or **Ctrl** + **S**	Save

Layout view mode shortcuts

Shortcut	Description
Ctrl + **B**	**Bold**
Ctrl + **I**	*Italic*
Ctrl + **U**	Underline
Ctrl + **E** + **H**	Hide selected elements on active screen (**E** indicates active editor).
Ctrl + **E** + **L**	Lock selected elements on active screen (**E** indicates active editor).

Script view mode shortcuts

Shortcut	Description
Ctrl + **Space**	Activate IntelliSense.
Tab	Complete word (IntelliSense).
Ctrl + **Shift** + **Space**	Parameter info.
Ctrl + **F**	Opens **Find** dialog.
Ctrl + **H**	Opens **Find and Replace** dialog.
F3	Find next.
Ctrl + **G**	Go to line.
Ctrl + **K**, **L**	List members.
Ctrl + **K**, **X**	Code snippets.
Ctrl + **K**, **C**	Comment selection.
Ctrl + **E**, **U**	Uncomment selection.

Configuration pages shortcuts

Shortcut	Description
Arrow keys	Move selection.
Tab	Move selection.
F4	Open selection list.
Enter	Confirm a selection.
Esc	Cancel a selection.
Delete	Remove a field or row.
Space	Toggle value.
Ctrl + **Tab**	Shifts focus away from the table.
↓ *(when at the last table row)*	Automatically fill fields.
* A **layered page** contains sub-tabs for further categorization.	

Shortcut	Description
Ctrl + ↓ *(only for layered pages*)*	Navigate into an expanded group; exit when positioned on the last row.
↑ *(only for layered pages*)*	Navigate up and out of a group.
Ctrl + **+** *(only for layered pages*)*	Expand a group.
Ctrl + **-** *(only for layered pages*)*	Collapse a group.
* A **layered page** contains sub-tabs for further categorization.	

4.2. Zoom and Pan

Use the following methods to **zoom** and **pan** when working in the in the Editor Area:

Zoom

- Select a zoom value from the **Zoom dropdown** in the Layout View Mode Controls.
- Adjust the zoom by moving the **Zoom slider** or clicking the **-** and **+** buttons in the Layout View Mode Controls..
- Place the mouse pointer anywhere on the screen and use the mouse **scroll wheel** to zoom in and out.

Pan

- Press and hold **Spacebar**, the mouse pointer will change to a hand. Click, hold, and drag with the mouse to pan the screen.

4.3. Grid Settings

There are two different methods to align objects on screens: **Snap objects to other objects** and **Snap objects to grid**. The grid settings are managed from the **Grid Settings** dialog. To open it, right-click on the current screen and select **Grid** from the context menu.

NOTE
Grid settings are global and will affect all screens.

Parameter		Description
Snap to	**Do not snap objects**	Disables object snapping.
	Snap objects to other objects	Displays snap lines when an object is moved, aiding positioning. Blue lines indicate alignment with objects, and a red horizontal line shows center alignment. **TIP** For vertical center alignment, select **Home** > **Format** > **Arrange** > **Position Objects** > **Align** > **Align Center**.
	Snap objects to grid	Aligns objects to grid points when moved. Disables **Snap Lines** for dragged objects.

Parameter		Description
Grid	**Spacing**	Defines the number of pixels between grid lines.
	Show grid on screen	Displays the grid on the screen.
	Grid style	Sets the grid style. Choose between **dots** or **lines**.

4.4. Tool Window Layout

The Tool Windows in iX Developer can be customized for better usability. They can be shown, hidden, or moved to different positions on the desktop. See the following chapters for more information.

4.4.1. Dock/Undock Tool Windows

To dock or undock Tool Windows, follow these steps:

Dock

1. Drag the tool window toward the center of the desktop.
2. A **guide diamond** appears, indicating docking options for the four edges of the desktop.
3. Move the mouse pointer toward the desired docking edge. An outline of the window appears in the target area.
4. Release the mouse button to dock the window.

Undock

- Drag the window heading to any position on the screen.

NOTE
Auto-Hide must be turned off before a tool window can be moved.

4.4.2. Manage Tabbed Tool Windows

You can dock Tool Windows into an existing groups of tools, creating tabbed tool windows.

Create tabbed tool windows

1. If the tool is already docked, double-click its heading to make it float.
2. Drag the tool to the center of the existing tool group. A **guide diamond** appears.
3. Drop the tool at the **center** of the guide diamond to add it as a tab in the group.

Releasing a tool window from a tabbed group

- Drag the tab of the tool window you want to release, and drop it on the desktop to undock the tool.

NOTE
Auto-Hide must be turned off before a tool window can be moved.

4.4.3. Auto-Hide

Auto-Hide allows for a maximized desktop view by minimizing Tool Windows along the edges when they are not in use. It can be toggled on or off by clicking the **Auto-Hide pushpin** icon on the window title bar.

- When Auto-Hide is enabled, tool windows expand to their original size when the pointer hovers over them.
- If a hidden tool window loses focus, it slides back to its tab at the edge of the desktop area.
- While hidden, the tool window's name and icon remain visible on a tab along the desktop edge.

5. Controllers

iX Developer HMI panels can be connected to various types of automation equipment, such as PLCs, servos, and drives. In this manual, the term controller refers to any connected equipment.

IMPORTANT
This section describes how to work with controllers in iX Developer. For detailed information on specific controllers, refer to the Driver Manual for the corresponding controller.

It is possible to link Tags to more than one controller at the same time, allowing the same application to work with different controllers. Controllers can be enabled or disabled in the iX App, and the app can have multiple controller connections. Additionally, the app can connect to an external OPC UA Server for extended functionality.

5.1. Add a Controller

When launching a new project, one of the initial tasks is selecting the controller brand and model to connect. This is easily done from The Wizard. A project can connect to more than one controller, see chapter Configure Multiple Controllers. To add a new controller to an existing project, follow these instructions:

1. In the Project Explorer Window, click **Tags** to open the Tags Configuration Page.
2. Select the **Controllers** tab.
3. Click **Add**.
4. Choose your preferred controller in the dialog and click **OK**.
5. If you need to change the controller settings, click **Settings**. Controller settings vary with the selected controller.

5.1.1. DEMO Controller

The **DEMO controller** is used to design and test a project directly on the development PC without connection to an external controller. You can add and use the **DEMO controller** just like a regular controller driver.

Predefined tags in the DEMO controller

Data type	Tag ID	Description
BIT, BOOL	M0 - M99	Digital tags
INT16, UINT16	D0 - D99	Analog tags

Predefined counters and digital tags

Data type	Tag ID	Description
BIT	M100 - M104	Toggles OFF and ON every 1000 ms.
INT16	C0 - C4	Counts from 0 to 100 with 1000 ms interval by default; min/max values can be defined.

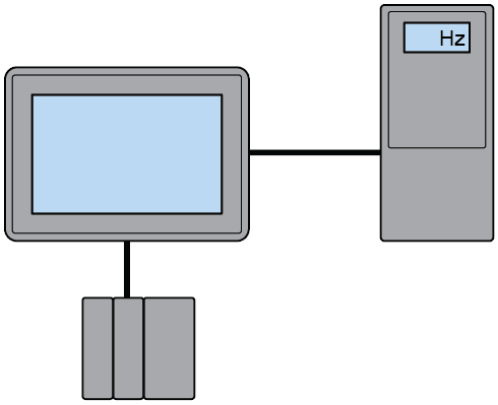
5.2. Configure Multiple Controllers

You can configure an HMI panel to communicate with multiple controllers simultaneously by using different drivers. This enables the HMI panel to interact with various controllers at the same time.

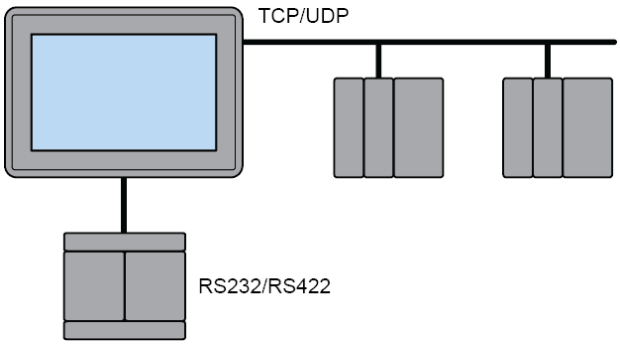
To connect multiple controllers and address tags:

- To connect a new controller to the project, see chapter Add a Controller.
- Follow the connection instructions provided in the Driver Manual.
- Address the tags in each controller according to the driver manual for that specific controller. When assigning a tag to an object, all tags, regardless of controller connection, are available for selection.

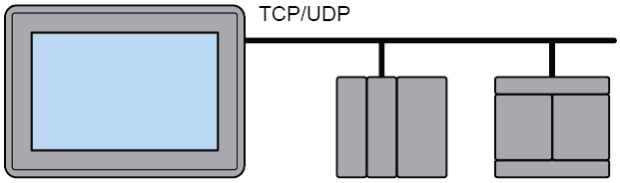
> **NOTE**
> If the same tag is linked to addresses in more than one active controller, it is not possible to control which controller's value is read in the iX App.



HMI panel communicating with one controller and one frequency converter



HMI panel communicating in series, with one controller via Ethernet, and with controllers of another make



HMI panel communicating with two controllers of different makes via Ethernet

5.3. Controller Drivers

IMPORTANT
This section describes how to work with controllers in iX Developer. For detailed information on specific controllers, refer to the Driver Manual for the corresponding controller.

5.3.1. Update or Install Drivers

Drivers can be updated via Internet or from a file. Each driver takes up approximately 3.0 MB.

5.3.1.1. Update or Install Drivers From the Internet

To update drivers using an internet connection, follow these steps:

NOTE
A web browser is not required.

1. Before updating the driver, determine whether the driver is of type **MPD** or **GEN2**:
   a. Go to **File** > **Update Drivers** > **Update Drivers From Internet**.
   b. In the list, check the **Driver type** column.
   Installation requirements
   - **MPD drivers**: Must be installed in administrator mode. iX Developer must be restarted after installation.
   - **GEN2 drivers**: Can be installed in user mode. No restart required.
2. *For **MPD drivers***: Right-click on the program icon and select **Run as administrator**.
3. Open the iX Developer project and go to **File** > **Update Drivers** > **Update Drivers From Internet**. The list shows the available drivers.
4. Select the driver you want to install or update. Use the **Mark Newer** option to automatically select all drivers that are newer than your installed versions, including drivers not yet installed.
5. Click **Download** to start the installation.
6. *For **MPD drivers***: Restart iX Developer.
7. Click **Project** > **Run** > **Rebuild** to rebuild the project.

5.3.1.2. Update or Install Drivers From a File

To update or install a driver from a file, follow the steps below. Note that only one driver can be processed at a time, which is useful for when reverting to an older driver.

1. Open iX Developer as administrator by **right-clicking** on the program icon and select **Run as administrator**.
2. Open the project and go to **File** > **Update Drivers** > **Update Drivers From File**.
3. Navigate to the folder where the driver files are located and click **Open** to display a list of installable drivers.

4. Select the driver you want to install or update. Use the **Mark Newer** option to automatically select all drivers that are newer than your installed versions, including drivers not yet installed.
5. Restart iX Developer as administrator by **right-clicking** on the program icon again and select **Run as administrator**.
6. Click **Project** > **Run** > **Rebuild** to rebuild the project.

5.3.2. Generic Driver Engine and GEN2 Drivers

The **Beijer Electronics Generic Driver Engine** allows iX Developer to use the **GEN2** type drivers. These drivers are smaller in size (around 250 kB) and can be installed in user mode without needing to restart the application.

The **Generic Driver Engine** is installed automatically with iX Developer and can be updated independently.

5.4. Synchronize the Controller Clock

To synchronize the controller clock with the iX Developer project, follow these steps:

1. In the Tags Configuration Page, click the small arrow next to the **Add** button and select **Add System Tag**.
2. Select the **Hour** system tag and click **OK**. This adds **SystemTagHour** to the tags list.
3. Add the **Minute** and **Second** system tags using the same procedure.
4. Link these tags to the corresponding addresses in the controller.
5. When any of these system tag values are updated, the changes will be sent to the controller.

5.5. Set Up Station Number for Controllers

For some controllers, you can specify the station from which values are read or written by assigning a station number as a prefix to the controller address.

Instructions

1. In the Tags Configuration Page, go to the **Controllers** tab.
2. Click **Settings**. This opens the **Controller Settings** dialog. Settings vary depending on controller type. The default station is addressed without a prefix.
3. Go to the **Stations** tab and define the stations. Stations can be defined as a **Fixed Number** or as a **Index Register**, see below.

   Fixed Number — With a fixed station number, an object is always linked to the same controller address. Only the value at this address can be accessed.

   Example

   If Station 2 is used, addressing `D10` in Station 2 would look like `2:D10`.

   Index Register — Using an index register allows dynamic station selection, enabling access to values from multiple controller stations with fewer tags. For more information on index registers, refer to Index Registers.

Benefits of using index registers for station handling

- **Reduces tag numbers** – One tag can dynamically point to different stations, reducing the total number of tags required.
- **Simplifies object configuration** – Index registers make it easier to configure objects, especially when accessing or modifying values across multiple stations.

4. Click **OK**.

6. Projects

This section describes how to work with projects in iX Developer.

TIP

- For basic information on how to create a new project using the **Wizard**, refer to Create a New Project With the Wizard.
- To ensure optimal performance, keep the items within the project to the Recommended Project Item Limits.

6.1. Project File Types

A project consists of two main sets of files:

- **Design Files**: These files relate to the functional and graphical design of the project.
- **App Files**: These files are compiled from the design files and are used in the iX App.

6.2. Project Folder

When a new project is created, a top-level folder named after the project is created, referred to as the **Project folder**. The files that define the functional and graphical design are stored at the top level of this folder. Additional folders are generated during validation and build processes.

NOTE

Do not save your iX project in a folder that syncs with **Microsoft OneDrive**, **Google Drive**, or similar services. This may prevent the project from opening and cause an **Access Denied** error.

6.3. Project Properties

The **Project Properties** includes iX App behavior, system font, and settings for mouse and keyboard. To open the project properties, click **Project** > **Settings**.

Properties
Advanced
Alarm Button
Database
Display/Target
Mouse/Touch Input Delay
Project Transfer
System Requirements
System Dialogs
Virtual Keyboard

6.3.1. Advanced

Parameter			Description
Screen Cache	**Set default value on all new screens to cache**		Enables the **Cached** option in **General** > **Screen** for all newly created screens. If unchecked, **Cached** is disabled by default for new screens. **NOTE** This does not apply to the start screen (always cached) and popup screens (never cached).
	Screen cache in project	**Set all to cache**	Enables **Cached** in **General** > **Screen** for all existing screens, except the start screen and popup screens.
		Remove all cache	Disables **Cached** in **General** > **Screen** for all existing screens, except the start screen and popup screens.
Filenames		**Enable FTP friendly names**	When enabled, file names with spaces are replaced with underscores, and periods (except for before the extension) are replaced with dashes. **Applies to** • **Generate Report**: Supported extensions: `.xls`, `.pdf`. For more information, see **Reporting** > **Generate Report** in Available Actions. • **Database Export** (**Data Logger Function**): For more information, see Exporting Database From the iX App. • **Database Export** (**Alarm Server Function**): For more information, see Exporting Database From the iX App. • **Database Export** (**Audit Trail Function**): For more information, see Export the Audit Trail Database From the iX App. • **Recipe Export**: For more information, see Export Recipe in iX Developer. • **Print Screen**: For more information, see **Screen** > **Print Screen** in Available Actions. **Example** File name changes from: `ReportTemplateName yyyy.mm.dd hh:mm:ss.extension` to: `ReportTemplateName_yyyy-mm-dd_hh-ss.extension`
OPC UA Structured Tags		**Enable Structured Tags**	Enables OPC UA Structured Tags in the project. Disabled by default.

Parameter	Description
Script Warnings Behavior	Determines how script warnings are handled during project build. **Options** • **Ignore warnings** - No warnings displayed. • **Show warning** - Warnings are shown. • **Treat warnings as errors** - Warnings are treated as errors.

6.3.2. Alarm Button

Parameter	Description
Show Screen	Specifies the screen that is displayed when the predefined alarm key on a keyboard-operated panel is pressed.

6.3.3. Database

Parameter		Description
Database location		Sets location for the active database. • **Hard drive** - Saves the database to the HMI panels internal flash drive. • **SD card** (default) - Saves the database to the SD card. Beijer Electronics recommend this option to reduce flash wear, see Use SD Card to Reduce Flash Wear.
Database Settings	**Back up database at startup**	When enabled, a backup of the database is created at startup if the integrity check is successful. If the integrity check fails, the system uses the previous backup. If no backup is available and the database is corrupt, the corrupt database will be replaced with a new, empty one. **NOTE** Enabling this option increases the HMI panels start-up time.
	Max size (MB)	When the database exceeds this size, the system tag **Database Max Size Exceeded** is set. For more information, see Available System Tags.

6.3.4. Display/Target

Parameter	Description
Window	• **No Title Bar**: Displays the project without title bar in the iX App. This alternative cannot be used for popup screens. • **Title Bar**: Displays the project with its title and a set of minimize, maximize and close buttons in the title bar. • **Close Button**: Displays the project with a title and a close button in the title bar.
	Maximize On Startup *(only for PC targets)*: When enabled, the app window is maximized in the iX App.
Screen Size *(only for PC targets)*	Defines the screen resolution for the project.
Target Rotate	Not available in the current version of iX Developer.
Target Type	Displays the selected target device. To change the project target, see Change Project Target.

6.3.5. Mouse/Touch Input Delay

Parameter	Description
Time (ms)	Configures the input delay for mouse and touch actions in the iX App. This delay can be applied globally to all objects that have **Delay Mouse/Touch Input** enabled without a specific delay, or it can be set individually for each object. To configure the delay for individual objects, use the Property Grid Window. To reset to the delay setting, enter **0**.

6.3.6. Project Transfer

Parameter	Description
Include source code when transferring project	When enabled, the software transfers a zipped file containing all source code to the HMI panel during App transfer. **IMPORTANT** This option must be enabled to use the Fetch the Project From the HMI Panel feature later.
Set password to zip file	Protects the zipped source code file with a password.

6.3.7. System Requirements

This chapter describes how to define the **System Requirements** in iX Developer.

In iX Developer 3, you cannot configure system settings directly as in previous versions. Instead, you define **System Requirements** that are evaluated and then reserved by the OS3 operating system. If the requirements cannot be fulfilled, the iX App will not start.

TIP

- For information on the HMI panels system settings, see App-Required System Settings in the **OS3 User Guide**.
- When converting an iX Developer project from **iX2** to **iX3**, you can manually import the system settings from **iX2** to the **System Requirements**. For instructions, see Migrate System Settings from iX2 to iX3 in the **X2 to X3 - Transition Guide**.

Parameter	Description
COM port *(only for HMI panel targets)*	Enter the required mode for a COM port. Select **Any**, **RS-232**, **RS-422**, **RS-485** or **CAN**. The number of available ports and possible configuration differ depending on selected target. For configuration instructions, refer to Serial Ports in the OS3 User Guide.
Network interface - Default gateway	Enter the required default gateway for the selected network interface.
Network interface - IP address	Enter the required IP address for the selected network interface.
Network interface - Primary DNS server	Enter the required primary DNS server for the selected network interface.
Network interface - Secondary DNS server	Enter the required secondary DNS server for the selected network interface.
Network interface - Subnet mask	Enter the required subnet mask for the selected network interface.
Power saving - Automatically turn off display	Select **Yes** to enable automatic backlight turn-off. This reduces backlight usage, helping extend the lifespan of HMI panels.
Power saving - Seconds until display is automatically turned off *(only available if **Power saving - Automatically turn off display** is set to **Yes**)*	Select the duration after which the backlight turns off automatically.
TCP server port	Enter the required TCP server port number for incoming traffic.
Time - SNTP synchronization	Select **Yes** if the panel clock needs to be synchronized with an external SNTP (Simple Network Time Protocol) server.
Time - SNTP server *(only available if **Time - SNTP synchronization** is set to **Yes**)*	Enter the SNTP server name for time synchronization. **NOTE** For cloud-compatible panels, it is recommended to use **time.windows.com**. A build warning will appear if SNTP time sync is not enabled or is not set to the recommended server. A warning message will also display when modifying these settings for cloud-compatible panels.

Parameter	Description
Time - SNTP update interval (only available if *(only available if **Time - SNTP synchronization** is set to **Yes**)*	Enter the synchronization update interval in minutes.
UDP server port	Enter the required UDP server port number.

6.3.8. System Dialogs

Parameter		Description
Color	**Form Color**	**Background**: Background color of system dialogs. **Foreground**: Font and foreground element color.
	Control Color	**Background**: Color of buttons and controls. **Foreground**: Font and icon color for buttons and controls.
Title Text	**Font**	Font used for dialog titles.
	Font Size	Font size for dialog titles.
General Text	**Font**	Font used for other text.
	Font SIze	Font size for other text.
MessageBox Timeout		Sets how long certain dialogs stay open without user interaction. Default is 20 seconds.
List Item Height		Sets the height of list items in system dialogs.

6.3.9. Virtual Keyboard

The virtual keyboard in the iX App is affected by settings in the **Multiple Languages** function. For more details, see Set Up Multiple Languages. The following settings apply when using the default language.

Parameter			Description
Enable Virtual Keyboard			Enables the virtual keyboard.
Default Keyboard Layout			Select the default keyboard layout from the following options: **Arabic, French, German, Hebrew, Italian, Korean, Russian, Simplified Chinese (Compact), Simplified Chinese, Spanish, Swedish, Traditional Chinese, US**.
Keyboard Size Customization	**Alpha Numeric Keyboard**	**Enable Keyboard Size Customization**	Check this option to customize the size of the alpha-numeric or numeric keyboard. Choose whether scaling is based on the Width or Height of the target resolution. Use the slider to set the keyboard size as a percentage of the target width or height.  **TIP** Test the keyboard size by simulating the iX App before transferring it to the target device. For the most accu-

Parameter			Description
	Numeric Keyboard		rate scaling, ensure the PC's DPI settings are set to 100%.

6.4. Build, Rebuild, Run or Simulate

The Project Ribbon Tab contains the following commands that enables building and simulation of the project. The table below highlights the differences between these commands:

Command	Validates the script code	Compiles the project	Deletes temporary files	Runs the iX App on the development PC	Communicates with the controller(s)
 Build	Yes	Yes Only changes are recompiled.	No	No	No
 Rebuild	Yes	Yes Entire project is recompiled.	Yes	No	No
 Run	Yes	Yes	Yes	Yes	Yes
 Simulate	Yes	Yes	Yes	Yes	No

NOTE

- If **project validation fails**, check that the absolute path to the project, including its components, does not exceed 260 characters.
- **Projects must be stored on the local hard drive** for iX Developer to be able to **Run** or **Simulate** them.
- When launching the **Build**, **Rebuild** and **Run** commands iX Developer will automatically check for **unused tags**. This function can be disabled or enabled from **CheckForUnusedTags** in iX Developer Options.

How to execute and quit each command

Command	Execute	Quit
 Build	Click **Build** in the Project Ribbon Tab.	
	Press **F6**.	
 Rebuild	Click **Rebuild** in the Project Ribbon Tab.	
 Run	Click **Run** in the Project Ribbon Tab.	Press **Alt** + **F4**.
	Click **Run** in the Quick Access Toolbar.	
	Press **F5**.	

Command	Execute	Quit
	Press **Alt** + **4**.	
 Simulate	Click **Simulate** in the Project Ribbon Tab.	Press **Alt** + **F4**.
	Click **Simulate** in the Quick Access Toolbar.	
	Press **Alt** + **5**.	

TIP

For more information about the commands, see Run Group.

6.5. Transfer, Fetch, Export and Import

iX Apps, projects and databases can be moved between devices using the following commands:

Transfer

Compiles and transfers the iX App from the development PC to the HMI panel.

Fetch Databases

Fetches the project databases from the HMI panel to the development PC.

Export

Compiles and saves the iX App on the development PC, allowing it to be imported to an HMI panel later.

Import

Imports the exported iX App to the HMI panel.

Fetch From Target

Fetches the entire iX project source code from the HMI panel to the development PC.

6.5.1. Transfer the iX App to the HMI Panel

To transfer the iX App to the HMI panel, follow the steps below or watch the video tutorial.

IMPORTANT

- All files and databases on the target HMI panels will be overwritten.
- To use the Fetch From Target command later, enable **Include source code when transferring project** in Project Properties > Project Transfer before the transfer.

VIDEO TUTORIAL

https://player.vimeo.com/video/1057814388?badge=0&autopause=0&player_id=0&app_id=58479

Instructions

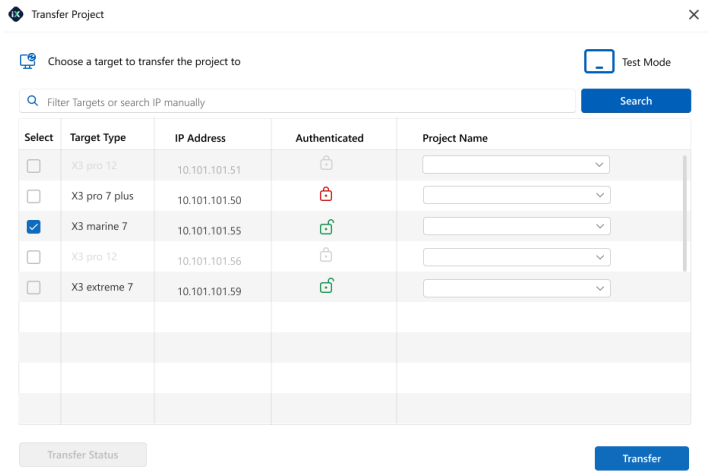
1. Connect the HMI panel to the development PC over the network.
2. In iX Developer, go to **Project** > **Transfer** > **Transfer**.
3. Select a **transfer mode**:

Test Mode	Faster build time, but slower runtime performance.
Production Mode	Slower build time, but optimized runtime performance.

TIP

Use **Test Mode** during development. Use **Production Mode** for final deployment.

4. In the **Transfer Project** client, all available HMI panels on the network with IP addresses and installed apps. Select the target panel and click **Transfer**.



5. For authentication, enter the login credentials for the OS3 user **Administrator** or **App Manager**. For more information see User Management in the **OS3 User Guide**.
6. Select **Accept Self-Signed Certificate** *(required only for the first transfer)*. See the Self-Signed Certificate chapter for details.
7. Click **OK**. Your authentication remains valid for **12 hours** unless you close iX Developer. The lock symbol under **Authenticated** indicates if you are authenticated.
8. A progress bar shows the status of the transfer. The panel stops the current app, downloads the new app, and automatically starts it.

NOTE

Time zone, region or daylight saving settings will not be transferred if these settings have been changed manually on the HMI panel.

6.5.2. Export the iX App

The **Export** feature compiles and saves the iX App on the development PC. This allows you to import it later to the HMI panel, see Import the iX App to the HMI Panel Using Transfer Client CLI for more details.

IMPORTANT
To use the Fetch From Target command later, enable **Include source code when transferring project** in Project Properties > Project Transfer before the transfer.

1. In iX Developer, go to **Project** > **Transfer** > **Export**.
2. Select a **transfer mode**:

Test Mode	Faster build time, but slower runtime performance.
Production Mode	Slower build time, but optimized runtime performance.

3. Choose a destination for the exported project and click **OK**. A progress bar will indicate the export status.

6.5.3. Import the iX App to the HMI Panel

You can import an exported iX App in two ways:

Transfer Client (simple)	Use this method to manually import the iX App to one HMI panel. See instructions below.
Transfer Client CLI (advanced)	Use this method to import the iX App to multiple HMI panels or when manual import is not practical. See Import the iX App to the HMI Panel Using Transfer Client CLI.

6.5.3.1. Import the iX App to the HMI Panel Using the Transfer Client

IMPORTANT
All files and databases on the target HMI panels will be overwritten.

Prerequisites

- Export the iX App beforehand using the **Export** command in iX Developer. See Export the iX App.

Instructions

1. Connect the HMI panel to the development PC over the network.
2. On the development PC, navigate to the `{Exported App}\TransferClient` folder.
3. Open `X3TransferClient.exe`.
4. In the **Transfer Project** client, all available HMI panels on the network with IP addresses and installed apps. Select the target panel and click **Transfer**.
5. For authentication, enter the login credentials for the OS3 user **Administrator** or **App Manager**. For more information see User Management in the **OS3 User Guide**.
6. Select **Accept Self-Signed Certificate** *(required only for the first transfer)*. See the Self-Signed Certificate chapter for details.

7. Click **OK**. Your authentication remains valid for **12 hours** unless you close iX Developer. The lock symbol under **Authenticated** indicates if you are authenticated.
8. A progress bar shows the status of the transfer. The panel stops the current app, downloads the new app, and automatically starts it.

> **NOTE**
> Time zone, region or daylight saving settings will not be transferred if these settings have been changed manually on the HMI panel.

6.5.4. Fetch the Project From the HMI Panel

The **Fetch From Target** command, located in the **File** menu, allows you to download the project source code from the HMI panel to the development PC.

> **IMPORTANT**
> The **Include source code when transferring project** option must have been enabled in Project Properties > Project Transfer when the iX App was transferred to the HMI panel.

> **TIP**
> To reduce project size before fetching the project, first run the **Clear Datalogger** action, then run the **Database Cleanup** action. For more information on actions, see the Actions chapter.

Instructions

1. Click **File** > **Fetch from Target**.
2. Follow the on-screen instructions to complete the process.

6.5.5. Transfer Logs

The transfer log tracks and displays the progress and details of data transfers between devices. Use the expand/collapse button in the **Transfer Client** to show or hide the log.

Transfer log message color codes

- **Green**: Success messages.
- **Red**: Failure, error, or exception messages.
- **Black**: General informational messages.

After a transfer completes, the log file is saved to: **%temp%\Transfer Logs**

Each transfer creates one log file per device, named as follows:
[TimeStamp]_[IPAddress]_[TargetType]_[TransferType].log

> **NOTE**
> If a project is transferred/imported to an HMI panel and settings have changed since the last transfer, the previous log on the panel is replaced.

6.6. Back Up the Project

To create a backup of an iX Developer project, follow these steps:

1. Go to **File** > **Back Up Project**, or **Project** > **Back Up Project**.
2. Choose the location to save the ZIP file.
3. *Optional*: Set a password to protect the ZIP file.
4. Click **OK** to complete the backup.

6.7. Change Project Target

To change the project target:

1. Click **Project** > **Settings** to open the Project Properties.
2. Select Display/Target.
3. Choose the **Target Type**.
4. *(For PC targets only)* Change the **Screen Size** if needed.

> **NOTE**
> **Target rotate** is not yet available for X3 HMI panels.

5. Click **OK**.
6. Select if you want to back up the project.

> **IMPORTANT**
> Beijer Electronics strongly recommend to make a backup of the project before changing the project target.

7. Click **Convert**.
8. If the new screen size differs from the old target, you can then choose to **automatically resize objects** to fit the new screen. Resizing adjusts horizontal and vertical scales independently, optimizing for wide-screen targets. This may change the aspect ratio of objects.
   Items resized automatically
   - Graphical objects
   - Popup screens
   - Font
   - Dynamics size and move settings

LIMITATIONS

- Some objects and styles may not be compatible with the new target and will be deleted. A log of removed objects is displayed after the transformation and can be saved as a file.
- Reverting to the previous target does not restore removed objects.

7. Transfer Client CLI

The **Transfer Client CLI** is a command-line tool used to import an iX App to one or more HMI panels, or to collect database or log files from an HMI panel. The tool runs without confirmation dialogs and is designed for use in automated deployment, remote updates, and script-based maintenance of HMI applications.

It is especially useful in production environments where manual deployment is not practical.

7.1. Program Locations

You can find `TransferClientCLI.exe` in the following locations:

- `C:\Program Files\Beijer Electronics AB\{iX Developer version}\Bin\TransferClientCLI`
- Included in exported iX App projects, in the `TransferClient` folder.

TIP

For export instructions, see Export the iX App.

7.2. Import the iX App to the HMI Panel Using Transfer Client CLI

Use the Transfer Client CLI to import the iX App to multiple HMI panels or when manual import is not practical. To manually import the iX App using the regular **Transfer Client**, follow the instructions in Import the iX App to the HMI Panel Using the Transfer Client.

You can run the program from the command prompt or use an Arguments File. The instructions below show how to use the command prompt.

NOTE

Arguments entered in the **command prompt** override argument in the **arguments file**.

IMPORTANT

All files and databases on the target HMI panels will be overwritten.

Prerequisites

- Install **.NET 8.0 Runtime** or later. Download it from the Microsoft website.
- Export the iX App beforehand using the **Export** command in iX Developer. See Export the iX App.

Instructions

1. Connect the HMI panel to the development PC over the network.
2. Open a command prompt on the development PC.

3. Navigate to the `{Exported App}\TransferClient` folder.
4. To import the iX App, enter the command:

   ```
   TransferClientCLI.exe -m Transfer -
   t {ipAddress1} {ipAddress2} ... {ipAddressN}
   ```

   Replace `{ipAddressX}` with the IP address(es) of the target panel(s).
5. If the port number or user credentials are needed, use this format:

   ```
   ipAddress[:port][,username][,password]
   ```

6. For more command options, see Arguments Overview.

TIP

Transfer logs are saved in a folder named **TransferLogs**. For more details, see Transfer Logs.

Example commands

Transfer to a single target	`TransferClientCLI.exe -m Transfer -t 192.168.98.1`
Transfer to multiple targets	`TransferClientCLI.exe -m Transfer -t 192.168.98.25 192.168.98.50`

7.3. Fetch Files from the HMI panel Using Transfer Client CLI

Use the Transfer Client CLI to collect databases and log files from the HMI panel without opening iX Developer.

You can run the program from the command prompt or use an Arguments File. The instructions below show how to use the command prompt.

NOTE

Arguments entered in the **command prompt** override argument in the **arguments file**.

Prerequisites

- Install **.NET 8.0 Runtime** or later. Download it from the Microsoft website.

Instructions

1. Connect the HMI panel to the development PC over the network.
2. Open a command prompt on the development PC.
3. Navigate to `/Program Files/Beijer Electronics AB/{iX Developer version}/Bin/TransferClientCLI`.
4. Build and run your command. See Arguments Overview for command syntax.

Example command

```
TransferClientCLI.exe -a true -o true -m fetch -s "c:\temp" -u a1b416f0-
db91-4ed1-8e62-c4bdb9d0a6cd -f "appdata://Database.db" -t 100.71.254.138
```

7.4. Arguments File

The file `transferclientcli.args` is included with an exported iX App, in the `TransferClient` folder. You can use this file to define arguments instead of entering them manually in the command prompt.

TIP

- To learn how to export the iX App, see Export the iX App.
- For a list of all available arguments, see Arguments Overview.

NOTE

Arguments entered in the **command prompt** override argument in the **arguments file**.

Example - Transfer Mode

```
{
  "Mode": "Transfer",
  "Targets": [
    {
      "IpAddress": "192.168.98.25",
      "Username": "iXuser1",
      "Port": 443
    }
  ],
  "AppPackages": [
    "..\\app.ulpkg"
  ],
  "Files": [],
  "AcceptSelfSigned": true,
  "SaveTo": "",
  "OverwriteFiles": true
}
```

Example - Fetch Mode

```
{
  "Mode": "Fetch",
  "Targets": [
```

```
    {
      "IpAddress": "192.168.98.25",
      "Username": "iXuser1",
      "Port": 443
    }
  ],
  "AppPackages": [],
  "Files": ["appdata://Database.db"],
  "AcceptSelfSigned": false,
  "SaveTo": "c:\\temp",
  "OverwriteFiles": false
}
```

7.4.1. Execute the Arguments File

To execute the arguments file:

1. Open a command prompt on the development PC.
2. Navigate to the `TransferClient` folder, see Program Locations.
3. Enter:

   ```
   TransferClientCLI.exe
   ```

7.5. Arguments Overview

Use the arguments below when running **Transfer Client CLI** from the command prompt or the Arguments File.

NOTE
Arguments entered in the **command prompt** override argument in the **arguments file**.

Argument list

Argument (command prompt)	Argument (arguments file)		Description
`-t`, `--targets`	`Targets`	`IpAddress`	Sets the IP configuration of the target HMI panel. **Command prompt format**: `ipAddress[:port][,username][,password]`. Separate multiple targets with spaces. **Arguments file**: Passwords are not allowed for security reasons.
		`Username`	
		`Port`	
`-a`, `--accept-self-signed`	`AcceptSelfSigned`		Accepts Self-Signed Certificate. Use `true`, `yes`, `y` to accept. Use `false`, `no`, `n` to reject.

Argument (command prompt)	Argument (arguments file)	Description
`-p, --packages`	`AppPackages`	Sets which application packages to transfer *(only for Transfer mode)*.
`-m, --mode`	`Mode`	Sets the operation mode. Use `Transfer` to transfer the iX App to the HMI panel. Use `Fetch` to collect database and log files from the HMI panel.
`-s, --saveto`	`SaveTo`	Sets the folder path for fetched files *(only for Fetch mode)*.
`-u, --uid`	*Not available*	Specifies the unique identifier (GUID) for the iX App. If unknown, open the Project folder and locate the file `{ProjectName}.neoproj`. Open it in a text editor and search for **GUID**.
`-f, --files`	`Files`	Lists files to fetch from the target *(only for Fetch mode)*.
`-o, --overwrite-files`	`OverwriteFiles`	Allows file overwrite on the target. Use `true`, `yes`, or `y` to overwrite without prompt. Use `false`, `no`, or `n` to cancel if files already exist.
`--help`	*Not available*	Shows help information.
`--version`	*Not available*	Shows version information.

Command prompt examples

Transfer:

```
TransferClientCLI.exe -t 192.168.1.25 192.168.1.113 -m transfer -p app.ulpkg
```

Fetch:

```
TransferClientCLI.exe -a true -o true -m fetch -s "c:\temp" -u a1b416f0-
db91-4ed1-8e62-c4bdb9d0a6cd -f "appdata://Database.db" -t 100.71.254.138
```

Arguments file examples

For argument file examples, see the Arguments File chapter.

8. User Management

This section explains how to configure user access in the iX App. It covers local user account setup, assignment to security groups, and domain login options such as Active Directory, Microsoft Entra ID, and ADFS. It also describes how to control access to screens and objects based on user roles and permissions.

By using passwords and security groups, you can build a security system for the project. You can assign different access levels to operators to limit their access to specific objects and functions. The project stores all security group data and passwords. Access levels are defined for each user group.

8.1. Configure User Accounts

You can configure user accounts in two ways:

- **From iX Developer** - Follow the instructions below.
- **From the iX App** - Open the **Users Dialog** by triggering the action **Show Users Dialog**. For more information on this action and how to add it to the project, see Available Actions and Add a New Action to the Action Menu Object.

NOTE
Users created in iX Developer is not possible to remove in the iX App.

Set up user accounts in iX Developer

1. Open the Security Configuration Page and select the **Users** tab.
2. Enter a **Username**.
3. Enter a **Password**.

IMPORTANT
Follow the Password Guidelines.

4. Enter a **Description** of the user (optional, not visible to the user).
5. Assign the user to one or more **Security Groups**. For details, see Define and Assign Security Groups.

8.2. Define and Assign Security Groups

To restrict access to specific Objects and Screens, assign them to a security group. Only users in the assigned group can control or view these elements.

Define a security group

1. Open the Security Configuration Page and select the **Groups** tab.
2. Click **Add** to create a new security group.
3. Enter a **Name** for the group.
4. Add **Users** to the group.

5. *(Optional)* Enable **Users Invisible in Runtime** to hide users in this group from the **Login** dialog in the iX App.

Assign a security group

- **To an object**: Select the object, then go to **General** > **Tag/Security** and select a security group. For more information, see Object Access and Visibility.
- **To a screen**: Select the screen in the Project Explorer Window, then go to **General** > **Tag/Security** and select a security group.

8.3. Domain Login

In the **Domain Login** tab of the Security Configuration Page you can configure domain login for the HMI panel users. The following domain login options are available:

8.3.1. Active Directory (AD)

AD is an on-premises directory service used to manage users, computers, and resources within a network. It provides centralized authentication and authorization through protocols like Kerberos and LDAP.

- **Use case**: Internal network management.
- **Setup instructions**: Set Up Active Directory.

8.3.2. Microsoft Entra ID

Microsoft Entra ID (formerly Azure AD) is a cloud-based identity management service for managing users across cloud and on-premises applications. It supports Single Sign-On (SSO), multi-factor authentication, and integration with third-party services.

- **Use case**: Cloud and hybrid environments.
- **Setup instructions**: Set Up Microsoft Entra ID.

8.3.3. Active Directory Federation Services (ADFS)

ADFS extends AD to provide Single Sign-On (SSO) across different domains and external applications. It uses federation protocols like SAML and OAuth to enable secure access to resources outside the organization.

- **Use case**: Secure external access and identity federation.
- **Setup instructions**: Set Up Active Directory Federation Services.

8.3.4. Set Up Active Directory

NOTE

For assistance with this configuration, contact your **IT representative** or **System Administrator**.

VIDEO TUTORIAL ON HOW TO SET UP ACTIVE DIRECTORY

https://player.vimeo.com/video/1058598463?badge=0&autopause=0&player_id=0&app_id=58479

Instructions

1. Open the Security Configuration Page.
2. Navigate to the **Domain Login** tab.
3. Under **Domain Login Type**, choose **Active Directory**.
4. Enter your **Domain**.
5. Click **Test Configuration** and enter login credentials for a user with access. Green checkmarks indicates a successful configuration.
6. Enable **Local Users** to allow login from both domain users and local users.

> **IMPORTANT**
> Domain login requires an Internet connection. If there is no Internet connection and **Local Users** are disabled, login will not be possible.

7. In **Domain Groups Mapping**, assign domain groups to match your local security groups.
8. Click **Test Configuration** and enter login credentials for a user with access. Green checkmarks indicates a successful configuration.

8.3.5. Set Up Microsoft Entra ID

> **NOTE**
> For assistance with this configuration, contact your **IT representative** or **System Administrator**.

1. Open the Security Configuration Page.
2. Navigate to the **Domain Login** tab.
3. Under **Domain Login Type**, choose **Microsoft Entra ID**.
4. Enter **Authority URI**, **Client ID** and **Redirect URI**.
5. Click **Test Configuration** and enter login credentials for a user with access. Green checkmarks indicates a successful configuration.
6. Enable **Local Users** to allow login from both domain users and local users.

> **IMPORTANT**
> Domain login requires an Internet connection. If there is no Internet connection and **Local Users** are disabled, login will not be possible.

7. In **Domain Groups Mapping**, assign domain groups to match your local security groups.
8. Click **Test Configuration** and enter login credentials for a user with access. Green checkmarks indicates a successful configuration.

8.3.6. Set Up Active Directory Federation Services

NOTE
For assistance with this configuration, contact your **IT representative** or **System Administrator**.

1. Open the Security Configuration Page.
2. Navigate to the **Domain Login** tab.
3. Under **Domain Login Type**, choose **Active Directory Federation Services**.
4. Enter **Authority URI**, **Client ID**, **Redirect URI**, **Scope** and **Group Claim**.
5. Click **Test Configuration** and enter login credentials for a user with access. Green checkmarks indicates a successful configuration.
6. Enable **Local Users** to allow login from both domain users and local users.

IMPORTANT
Domain login requires an Internet connection. If there is no Internet connection and **Local Users** are disabled, login will not be possible.

7. In **Domain Groups Mapping**, assign domain groups to match your local security groups.
8. Click **Test Configuration** and enter login credentials for a user with access. Green checkmarks indicates a successful configuration.

8.4. User Access

You can assign screen and object access to specific users or user groups.

- To manage screen access, see Screen Access.
- To manage object access, see Object Access and Visibility.

9. Screens

In iX Developer, screens represent the visual interface of the project. Each screen is a workspace where you design the user interface in the iX App. Screens can include elements such as buttons, text fields, images, and other graphical components. You can create multiple screens for different functions or views. Each screen is linked to specific actions or events, allowing users to interact with the system.

TIP
For instructions on how to optimize screen usage, see Screen Performance.

9.1. Add a Screen to the Project

You can add screens to your project in multiple ways:

From the Home Ribbon Tab

- Click **Add Screen**.

From the Insert Ribbon Tab

- Click **Add Screen**.

From the Project Explorer Window

- Right-click on **screens** and select **Add Screen**.

From the Navigation Manager Window

- Use the mouse pointer to drag from one screen to an empty location.

9.2. Set a Screen Relation

Use screen relations to connect screens and enable navigation between them.

Instructions

1. Open the Navigation Manager Window: go to **View** > **Navigation Manager**.
2. Drag from one screen to another to create a relation.
3. Use Objects and Actions to define how users navigate between the screens.

NOTE
If the second screen is deleted, you must manually update or remove the button's action.

Example: Open another screen with a button

1. In the Navigation Manager Window, drag from the first screen to the second screen.
2. On the first screen, create a Button Object.

3. Go to the Actions Ribbon Tab.
4. In the **Select Action** dropdown, choose **Show Screen**.
5. In the screen list below, select the second screen.

When the button is clicked in the iX App, the second screen opens.

9.3. Change Screen Name, Screen Title or Screen ID

Each screen includes the following:

- **Screen Name** – The screen's name in iX Developer.
- **Screen Title** – The title shown in a popup in the iX App.
- **Screen ID** – A number used to identify or switch screens using a tag.

Change Screen Name

1. Select the screen.
2. Go to **Home** > **Name**.
3. Enter the new name.

Change Screen Title

1. Select the screen.
2. Go to **General** > **Screen** > **Screen title**.
3. Enter the new title.

Change Screen ID

1. Select the screen.
2. Go to **General** > **Screen** > **Screen ID**.
3. Enter the new ID.

9.4. Screen Roles

All screens have the same basic functions by default, but a screen can be assigned a specific role to specialize its behavior in the project.

Available screen roles

- **Startup screen**: The first screen displayed in the iX App. By default, **Screen1** is the startup screen. See Set Startup Screen.
- **Background and Foreground screen**: Any screen, except those with Aliases, can be used as a background or foreground screen by other screens in the project. See Set a Screen as Background or Foreground.
- **Screen template**: A screen saved as a template can be reused in the current project and in future projects. See Manage Screen Templates.
- **Popup screen**: A popup screen can be configured to behave like a floating window. Any screen can be set up as a popup. See Configure a Popup Screen.

9.5. Set Startup Screen

The startup screen is the first screen displayed when the project runs. By default, **Screen1** is set as the startup screen. The selected startup screen will be highlighted with a green outline in **Project Explorer** > **Screens**.

Set Startup Screen

1. Right-click on the desired screen in the Editor Area or in **Project Explorer** > **Screens**.
2. Select **Set as Startup Screen**.

9.6. Set a Screen as Background or Foreground

Use background or foreground screens to apply shared design elements across multiple screens. This helps create a consistent layout and reduces setup time.

NOTE
Screens with Aliases, cannot be used as a background or foreground screen.

Instructions

1. Open the screen in the Editor Area.
2. Go to **Home** > **Screen**.
3. In the **Parent Screen** dropdown, select the parent screen.
4. Choose **Background** or **Foreground**,.
5. Click **OK**.

The selected screen is now applied as a background or foreground. Any changes made to it will automatically update all screens that use it.

9.7. Manage Screen Templates

iX Developer includes several predefined screen templates with basic functionality.

Add a screen from screen templates

1. Go to **Home** > **Screen**.
2. Click the down arrow on the **Add Screen** button.
3. Select a template from the list.
4. If the template includes tags not in your project, choose whether to add them automatically.

Save screen as screen template

1. In the Project Explorer Window go to the Screens Section.
2. Right-click the screen and select **Save Screen as Template**.

Delete screen template

IMPORTANT
This will permanently delete the screen template folder from your disk.

1. Go to **Home** > **Screen**.
2. Click the down arrow on the **Add Screen** button.
3. Right-click the screen template you want to remove and select **Delete Screen Template**.

9.8. Manage Screen Groups

Use screen groups to keep your project organized and simplify screen management.

View screen groups

1. In the Project Explorer Window go to the Screens Section.
2. Double-click **All Screens** to view all available groups.

TIP
Screens not assigned to a group appear under **Ungrouped Screens**.

Create a screen group

1. In the Project Explorer Window go to the Screens Section.
2. Right-click **All Screens**.
3. Select **Add Group** and enter a unique name.

Assign a screen to the group

- **Option 1:**
  1. In the Project Explorer Window go to the Screens Section.
  2. Right-click the screen and select **Add to Screen Group**.
  3. Choose the group and click **OK**.

- **Option 2:**
  1. In the Project Explorer Window go to the Screens Section.
  2. Double-click **All Screens**.
  3. Drag and drop the screen onto the screen group.

Rename screen group

1. In the Project Explorer Window go to the Screens Section.
2. Double-click **All Screens**.
3. Right-click the group, and select **Rename Group**.

Delete screen group

IMPORTANT
Deleting a group also deletes all screens in that group.

1. In the Project Explorer Window go to the Screens Section.

2. Double-click **All Screens**.
3. Right-click the group, and select **Delete Group**.

9.9. Configure a Popup Screen

In the iX App, only one screen is usually visible at a time. You can configure any screen to open as a popup window.

Set a Screen as Popup

1. Go to **General** > **Screen** and select **Popup Screen**.
2. Enter the **start position coordinates (X, Y)** to define where the popup will open. Position (1,1) is the top-left corner of the screen.
3. Define the **width** and **height** in pixels.
4. Select **Modify Screen Size/Position** to adjust the popup screen by dragging it with the cursor. Coordinates update automatically.
5. Check **Modal** to prevent interaction with other screens while the popup is open. Background screens continue to update.
   Modal recommendations
   - It is not recommended to have more than two modal popup screens open simultaneously.
   - Avoid triggering multiple modal popups from background events like alarms or tag value changes.

Open and close the popup

- To be able to **open** the popup in the iX App, configure an object with a **Show Screen** action. For details, see Add an Action to an Object and Available Actions.

NOTE
The **Show Screen** action position settings override the screen's **General** > **Screen** settings.

- To be able to **close** the popup in the iX App, configure an object with a **Close Screen** action. For details, see Add an Action to an Object and Available Actions.

NOTE
If triggered by a tag the **Close Screen** action closes both the popup and its parent screen.

NOTE
When using a Web Browser Object in a popup screen, only the default style for the popup screen is supported.

9.10. Import a Screen from Another Project

Do this to reuse screens from other iX Developer projects.

Instructions

1. In the Project Explorer Window, go to the **Screens** section.
2. Right-click the **Screens** heading and select **Import**.
3. Select the screen you want to import and click **OK**.

Import considerations

- **Screen name conflicts**: If the imported screen's name matches an existing screen, a new name will be suggested.
- **Tag management**: If the imported screen contains tags that are not present in the current project, you will be prompted to add them. Only tag names will be imported; no additional tag information will be transferred.

9.11. Screen Access

Use screen security settings to restrict access based on user groups.

TIP

For more information on users and user groups, refer to the User Management chapter.

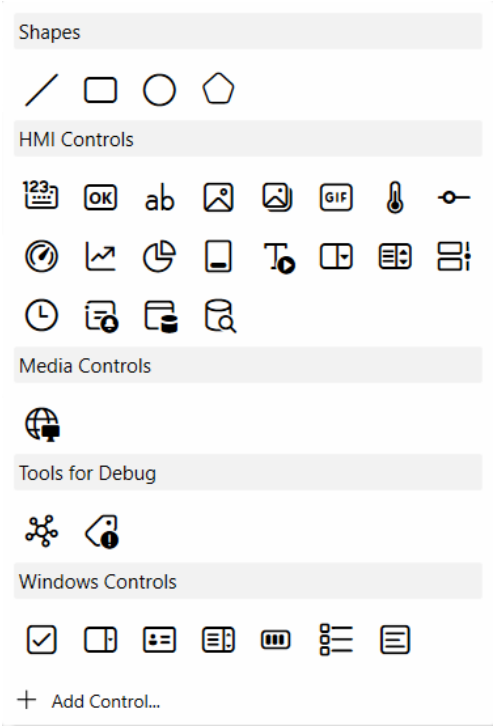
Configure screen access

1. Open the screen in the Editor Area.
2. Go to **General** > **Tag/Security**.
3. Select the required security group(s).

NOTE

- **Security overrides dynamics**: Security settings override dynamics.
- **Startup screen**: Security settings cannot be applied to the startup screen.
- **Background screens**: The security of the top-level screen always applies to the background screen.
- **Imported Screens**: These retain their original security settings from the source project.

10. Objects



In iX Developer, **objects** are fundamental elements used to create interactive and visually appealing user interfaces. Objects represent various components, such as buttons, text fields, images, charts, and other graphical elements that users interact with on the screen. Each object is designed to serve a specific function, contributing to the overall functionality and usability of the application.

Objects can be customized through various properties, enabling developers to modify their appearance and behavior according to the application's requirements. These properties include dimensions, colors, fonts, and actions triggered by user interactions, such as clicks or inputs. iX Developer provides an intuitive drag-and-drop interface, allowing developers to easily place and configure objects within the project workspace.

Furthermore, objects are hierarchical, meaning they can contain other objects, forming a structured layout that enhances the application's design. This hierarchical organization aids in the management of complex interfaces, making it easier to maintain and update the project.

This section will explore the different types of objects available, their properties, and best practices for their implementation in your projects.

10.1. Object Definitions and Descriptions

This section describe all objects available in the Home Ribbon Tab.

10.1.1. Shapes Objects

Shapes objects are versatile tools that can represent various geometrical forms with the ability to modify properties such as size, color, and border styles, allowing for the customization of screen designs to meet specific project requirements. Shapes objects enhance the visual hierarchy of user interfaces, guiding the users attention to key elements and information. They can also serve functional purposes, such as indicating status, creating buttons, or delineating sections of the interface.

Shapes objects

Symbol	Name	Instructions
	**Ellipse Object**	To maintain the aspect ratio while drawing the object, hold **Shift**.
	**Line Object**	To constrain a line object to 45-degree angles, hold **Shift** while drawing the line.
	**Poly Line Object**	To edit points on an existing Poly Line, **right-click** on the Poly Line and select **Edit Points**.
	**Rectangle Object**	To maintain the aspect ratio while drawing the object, hold **Shift**.

TIP

To change the appearance of shapes objects based on tag value changes, link tags to shapes using the controls in the Dynamics Ribbon Tab.

10.1.2. HMI Controls Objects

HMI Controls objects are used to facilitate communication between users and systems, enabling intuitive interaction through various input and output mechanisms.

HMI Controls objects

Symbol	Name
	Action Menu Object
	Alarm Viewer Object
	Analog Numeric Object
	Animated GIF Object
	Animated Label Object
	Audit Trail Viewer Object
	Button Object
	Chart Object
	Circular Meter Object
	Database Viewer Object

Symbol	Name
	Digital Clock Object
	Linear Meter Object
	Multi Picture Object
	Picture Object
	Roller Panel Object
	Slider Object
	Text Object
	Touch Combo Box Object
	Touch List Box Object
	Trend Viewer Object

10.1.2.1. Action Menu Object

The **Action Menu** is used to display multiple menu groups, containing different sets of actions triggered in the iX App from the action menu object.

To configure object settings, select the object and go to the General Ribbon Tab. The following settings are available:

Parameter		Description
Settings	**Edit Actions**	Opens the Edit Actions Window.
	View type	Sets the menu display style: **Bar**, **Expand** or **Tab**.
	Item Width *(only available for **View type**: **Bar** and **Expand**)*	Sets the width of the menu items.
	Item Height	Sets the height of the menu items.
	Orientation *(only available for **View type**: **Bar**)*	Sets the orientation of the menu items.
	Picture Width	Sets the width for images associated with the menu items.
	Picture Height	Sets the height for images associated with the menu items.

Parameter		Description
	Scroll Bar Width	Sets the width of the scroll bar.
Touch Settings	**Scroll Sensitivity**	Adjusts the scroll sensitivity of the menu. The minimum value is 1, and the maximum is 10.

TIP

To set tag dependent values for the object settings, link tags to the object using the controls in the Dynamics Ribbon Tab. This allows for dynamic updates based on tag values, enhancing the interactivity and functionality of the object.

Edit Actions Window

The **Edit Actions** window allows users to define various attributes and behaviors for each menu item in the Action Menu Object and its associated groups.

Settings

Parameter		Description
Actions	**Text**	The text to be displayed for the menu item.
	Picture	The picture displayed with the menu item.
	Make Transparent	Enables transparency for the menu item.
	Action	Specifies the action that occurs when the item is clicked, such as navigating to another screen or executing a command.
Menu Groups	**Menu Group Name**	The name of the menu group.
	Picture	The picture displayed for the menu group.
	Make Transparent	Enables transparency for the menu group.

NOTE

Avoid using semi-transparent images as HMI panels not fully support opacity.

Add a New Action to the Action Menu Object

Instructions

1. Select the desired **Action Menu Object**.
2. Click **General** > **Settings** > **Edit Actions** to open the Edit Actions Window.
3. In the **Actions** tab, click **Add** to insert a new item at the end of the list.
4. Select **Text** and **Picture** for the action.
5. *(Optional)* Check **Make Transparent** to make the picture transparent.
6. In the **Action** column, click the **...** button to open the **Actions Properties**.

7. Select the action to be displayed in the list and click **OK**.
8. Click **OK**.

Add a New Menu Group to the Action Menu Object

Instructions

1. Select the desired **Action Menu Object**.
2. Click **General** > **Settings** > **Edit Actions** to open the Edit Actions Window.
3. In the **Menu Groups** tab, click **Add** to add a new menu group to the list.
4. Enter the **Menu Group Name**.
5. Select **Picture** for the menu group.
6. *(Optional)* Check **Make Transparent** to make the picture transparent.

The menu group will now be displayed in the **Actions** tab as a separate menu group tab.

10.1.2.2. Alarm Viewer Object

The **Alarm Viewer** displays and manages alarm items, such as out-of-bounds tag values.

TIP
For more information the alarm viewer, refer to Alarm Viewer chapter.

To configure object settings, select the object and go to the General Ribbon Tab. The following settings are available:

Parameter		Description
Button	**Position**	Controls which borders the buttons align with.
	Acknowledge Selected	Adds the **Ack Selected** button that acknowledges selected alarms.
	Acknowledge All	Adds the **Ack All** button that acknowledges all alarms.
	Acknowledge Visible	Adds the **Ack Visible** button that acknowledges only visible alarms.
	Clear	Adds the **Clear** button that removes all alarms.
	Clear Visible	Adds the **Clear Visible** button that removes all visible alarms.
	Filter *(not supported in keyboard operated panels)*	Adds the **Filter** button that opens the filter configuration window.
	Info	Adds the **Info** button that performs the action set by **Alarm Event Info Requested** in Alarm Server Action Triggers.
	Play/Pause	Adds the **Play/Pause** button that starts and pauses the selected Alarm Viewer.

Parameter		Description
	Enable Context Menu	Shows the alarm viewer commands in context menu in the iX App.
	Enable/Disable Selected Alarm	Adds the **Enable/Disable selected** button that toggles the enable status of the selected alarm.
	Manage Alarms	Adds the **Manage Alarms** button that opens the Manage Alarms Popup Window.
Display Settings	**Configure Columns**	Selects and configures columns displayed in the iX App.
	Filter Settings	Filters alarm statuses to displayed in the iX App.
	Max Number of Rows	Sets the maximum number of alarms to display. If exceeded, scroll bars will appear and the oldest items will be deleted.
	Show Column Headers	Shows or hides the column headers in the Alarm Viewer.
	Show Summary	Displays status information at the bottom of the Alarm Viewer.

10.1.2.3. Analog Numeric Object

The **Analog Numeric** object displays a box with a numeric or alphanumeric value. The value can be connected to a tag.

To configure object settings, select the object and go to the General Ribbon Tab. The following settings are available:

Parameter		Description
Display Format	**Type**	Select **Integer**, **Decimal**, **String**, **Hex** or **Binary** display format.
	Limit Characters to	Defines the maximum number of displayed characters, represented by # symbols. Not supported for **Traditional Chinese, Simplified Chinese**, or **Simplified Chinese (Compact)**.
	Number of Decimals *(only for Type: Decimal)*	Defines the maximum number of displayed decimals, represented by the **#** symbol on the object.
	Zero Fill	Fills empty character positions with zeros to display the maximum number of characters.
Prefix/ Suffix	**Prefix**	Adds text before (Prefix) or after (Suffix) the object value.
	Suffix	
Tag/Security	**Select Tag**	Connects the object to a tag from the Tags Configuration Page.
	Select Security Groups	Restricts access based on security groups. If empty, access is unrestricted.

Parameter		Description
	Select Visibility	• **Default**: Inherits settings from the Security Properties Dialog. • **Disabled**: The object is visible but grayed out for unauthorized users. • **Hidden**: The object is invisible for unauthorized users. • **Normal**: The object appears normally but cannot be interacted with by unauthorized users.
Format	**Auto Stretch Font**	Resizes font to fit the object's height. To adjust manually, disable this function and use **Home** > **Font**.
	Disable Operator Input	Disables operator input in the iX App.
	Transparent	Makes the object's frame and background color invisible.
Text Alignment		Positions the text within the box.
Validation	**On Input**	Validates input values based on **Lower Limit** and **Upper Limit**.
	On Display	Changes background color according to **Home** > **Format** > **Other Colors**, based on **Lower Limit** and **Upper Limit** values.
	Lower Limit/ Upper Limit	Defines acceptable value ranges for **On Input** and **On Display** validation.

10.1.2.4. Animated GIF Object

The **Animated GIF** object includes a video in the project. Imported animated GIFs are added to the **Project Picture** folder in the Component Library Window.

The object can be connected to a tag that controls animation. When the value of the connected tag is 0, the animation is turned off and the picture is displayed static in its initial design. An object not connected to a tag loops the animation.

NOTE
Avoid using semi-transparent images as HMI panels not fully support opacity.

To configure object settings, select the object and go to the General Ribbon Tab. The following settings are available:

Parameter		Description
Settings	**Change Picture**	Opens a browse dialog for selecting a new picture.
	Reset Picture	Restores the picture's original height and width.
	Stretch	Adapts the picture's height and width to fit the selection frame when resizing.
	Transparent	Makes the object's frame and background color invisible.
Tag/Security	**Select Tag**	Connects the object to a tag from the Tags Configuration Page.

Parameter		Description
	Select Security Groups	Restricts access based on security groups. If empty, access is unrestricted.
	Select Visibility	• **Default**: Inherits settings from the Security Properties Dialog. • **Disabled**: The object is visible but grayed out for unauthorized users. • **Hidden**: The object is invisible for unauthorized users. • **Normal**: The object appears normally but cannot be interacted with by unauthorized users.

TIP

To set tag dependent values for the object settings, link tags to the object using the controls in the Dynamics Ribbon Tab. This allows for dynamic updates based on tag values, enhancing the interactivity and functionality of the object.

10.1.2.5. Animated Label Object

The **Animated Label** displays an animated text label in the iX App.

To configure object settings, select the object and go to the General Ribbon Tab. The following settings are available:

Parameter	Description
Text	Specifies the text displayed in the animated label.
Animated	Enables or disables animation for the label.
Animation speed	Controls the speed of the animation. (Min: 1, Max: 32 767).
Animation Direction	Defines the direction of the animation movement.

TIP

To set tag dependent values for the object settings, link tags to the object using the controls in the Dynamics Ribbon Tab. This allows for dynamic updates based on tag values, enhancing the interactivity and functionality of the object.

LIMITATIONS

- Avoid using braces **{}** around digits in text objects.
- Use this object with consideration, as it slightly increases CPU load.

10.1.2.6. Audit Trail Viewer Object

The **Audit Trail Viewer** displays information about operator changes that have been logged with the Audit Trail function. For detailed information about **Audit Trail**, refer to the Audit Trail Function chapter.

10.1.2.7. Button Object

The **Button** object can display text and/or a picture. If a picture is used, the picture is added to the **Project Pictures** folder in the Component Library Window. Buttons can be configured to perform specific tasks in the iX App via Actions or Scripts. To set the text to depend directly on a tag value, go to **Dynamics** > **General** > **General.**

To configure object settings, select the object and go to the General Ribbon Tab. The following settings are available:

Parameter		Description
Picture	**Picture**	Selects a picture to display on the button. Pictures from the Project Pictures folder is available for selection directly. To remove a previously selected picture, select **No Picture**. **NOTE** Avoid using semi-transparent images as HMI panels not fully support opacity.
	Multi Picture	Display multi pictures on the button. Refer to Configure Multi Pictures for a Button Object for configuration.
	Transparent	Makes the picture transparent.
Picture Alignment		Adjusts the picture alignment. Additional settings are available by clicking the arrow in the lower right corner.
Style		Select predefined graphical styles for the button. Additional predefined styles are found in the Configure Texts Window. **NOTE** Editing default button style properties (e.g., corner radius) may result in appearance differences between iX Developer and the iX App.
Text	**Text**	Sets the text displayed on the object. **NOTE** Avoid using braces **{}** around digits in text objects.
	Multiline	Allows text to run over multiple lines using **Enter**. Confirm the text entry with **Ctrl** + **Enter**.
	Word Wrap	Wraps the text to fit within the button's width.

Parameter		Description
	Configure Texts	Opens the Configure Texts Window to link the text to a tag.
Text Alignment		Adjusts the text horizontally or vertically.
Visibility	**Transparent**	Makes the object's frame and background color invisible.

Configure Texts Window

The **Configure Texts** window allows you to set the following properties:

Parameter	Description
Select tag	Connects the texts to a specific tag.
Add	Adds a new text entry to the list.
Delete	Removes a text entry from the list.
Text	The text that will be displayed. **NOTE** Avoid using braces **{}** around digits in text objects.
Start Value	The minimum and maximum values at which the text is displayed. If the tag value falls outside the defined range, the text field value is displayed.
End Value	

How to Use the Button Object

Create a Button with Momentary Function

A button with a momentary function activates only while being pressed and immediately deactivates when released.

Instructions

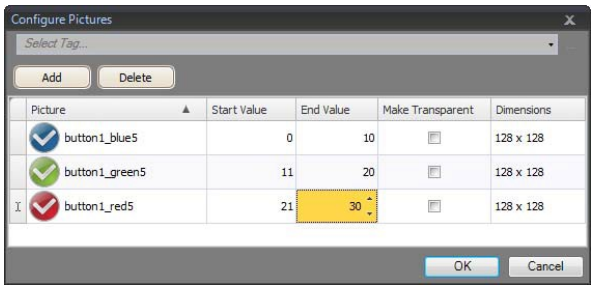
1. Select the Button Object.
2. Go to the Actions Ribbon Tab.
3. Select the **Set Tag** action for **Mouse Down**, and the **Reset Tag** action for **Mouse Up**.

Configure Multi Pictures for a Button Object

You can display different pictures on a button based on the value of a connected tag.

Instructions

1. Select the Button Object.
2. Go to **General** > **Picture**, and enable **Multi Picture**
3. Click **General** > **Picture** > **Picture**.
4. Select a tag to control which picture will be displayed.
5. Click **Add** to add pictures and set the start and end values for each picture.



6. Click **OK**.

The button will now display the corresponding picture based on the selected tag's value. If the value falls outside the defined range, no picture will be shown.

10.1.2.8. Chart Object

The **Chart** object is used to display values from connected array tags.

To configure object settings, select the object and go to the General Ribbon Tab. The following settings are available:

Parameter		Description
Chart Settings	**Edit Series**	Opens the Chart Series Window.
	Chart Types	• Area • Area Horizontal • Bar • Donut • Line • Line Horizontal • Pie • Points • Radar
	Header	Displays the name above the chart.
X/Y1/Y2 Axis Settings	**Visible**	Sets the visibility of the X/Y1/Y2 axis.
	Grid Visible	Sets the visibility of the X/Y1/Y2 axis grid lines.
	Automatic	Automatically calculates the minimum and maximum X/Y1/Y2 axis values. Overrides manual settings for Minimum and Maximum.
	Minimum	Sets the minimum value for the X/Y1/Y2 axis. Ignored if **Automatic** is enabled.
	Maximum	Sets the maximum value for the X/Y1/Y2 axis. Ignored if **Automatic** is enabled.
Style	**Style**	Sets the visual style of the chart.
Visibility	**Scale**	Sets the visibility of the scale.
	Transparent	Makes the chart transparent.

Parameter		Description
	Header	Sets the visibility of the chart's header.
	Legend	Sets the visibility of the chart's legend.

Chart Series Window

To open the **Chart Series** window:

1. Select the chart object.
2. Go to **General** > **Chart Settings**, and click **Edit Series**.

Parameter	Description
Name	The name of the chart series, as shown in the chart legend.
Value Array/Value Array 1	The tag array that provides values to the data series of the chart.
Value Array 2 *(only available for chart types: Area, Area Horizontal, Bar, Line, Line Horizontal, and Points)*	• If no value is set for **Value Array 2**: **Value Array 1** will provide the chart's values (e.g., determining the height of bars in a bar chart). • If Value **Array 2** is set: **Value Array 2** will provide the chart's values (e.g., determining the height of bars in a bar chart), while **Value Array 1** will provide the X-axis labels.
Size	Overrides the size of X and Y tag arrays. If not set, the smallest array size is used.
Axis/X-Axis/Slice Labels	Sets the label for each connected tag. The maximum is 256 rows. Axis labels are used for **Radar** charts, Slice Labels for **Pie** and **Donut** charts, and X-Axis labels for other chart types.
Color/Colors *(available for all chart types except Pie and Donut)*	Sets the color(s) of the series.
Thickness	Sets the thickness of the series, applicable for Line and Line Horizontal chart types.
Axis Type	Selects the Y-axis that the series will use.
Show Value Markers/Show Slice Labels	**Show Value Markers** displays markers near the tag values inside the chart. For **Pie** and **Donut** charts, this is called **Show Slice Labels**.

NOTE
Large arrays with frequently changing values may impact performance.

TIP
For information on how to set up **Array Tags**, refer to Set up Array Tags.

10.1.2.9. Circular Meter Object

The **Circular Meter** represents and displays a numeric value, connected to a tag. The graphical profile can be customized, with options for full, half, or quarter meters in different styles.

To configure object settings, select the object and go to the General Ribbon Tab. The following settings are available:

Parameter		Description
Angular Settings	**Start Angle**	Determines the placement of the scale on the meter. The angular value is based on a start (0 degrees) at three o'clock, in a counter clockwise direction from there. **NOTE** The start angle must differ from the end angle.
	End Angle	
	Text Rotation	• **Parallel**: Lets the numbers follow the outline of the meter. • **None**: Displays the numbers aligned horizontally.
	Clockwise	Sets the rotation direction, clockwise or counter clockwise.
Region Settings		Define colors and at which values to start and stop for up to 3 regions. Enter 0 for both **Min Value** and **Max Value** to display no region color.
Value Scale	**Min Value**	Start and end values of the scale. Values can include one decimal digit.
	Max Value	
	Major Ticks	Number of labeled scale marks.
	Minor Ticks	Number of ticks between two adjacent major ticks.
Style		Predefined graphical styles for the object. The smaller styles are intended to allow full visibility in the smaller HMI panels. Additional predefined styles are found in the Object Variations Group.
Needle Settings	**Analog Animation**	Enables smooth needle movement at value changes. When selected, it is possible to activate **Property Grid** > **Appearance** > **Shortest Path**. This setting affects the direction of the animated needle, resulting in a counter clockwise movement if this is the shortest path between two indicated values. **NOTE** Using animations can increase CPU load, so limit needle animation to a few meters per screen, especially on smaller panels.
	Oscillating Needle *(only available in combination with **Analog Animation**)*	Enables an oscillation effect of the needle around the new value.

Parameter		Description
Tag/ Security	**Select Tag**	Connects the object to a tag from the Tags Configuration Page.
	Select Security Groups	Restricts access based on security groups. If empty, access is unrestricted.
	Select Visibility	• **Default**: Inherits settings from the Security Properties Dialog. • **Disabled**: The object is visible but grayed out for unauthorized users. • **Hidden**: The object is invisible for unauthorized users. • **Normal**: The object appears normally but cannot be interacted with by unauthorized users.
Visibility	**Scale**	Activates scale color. Select scale color from **Home** > **Format** > **Other Colors**. When disabled, scale color is transparent
	Transparent	Makes the object's frame and background color invisible.

10.1.2.10. Database Viewer Object

The **Database Viewer** displays database contents in the iX App. Only databases included in the current project are accessible.

To configure object settings, select the object and go to the General Ribbon Tab. The following settings are available:

Parameter		Description
Settings	**Advanced**	Enables the **Database** and **Browse Data Table** parameters, as described below.
	Show Controls	Allows database and table selection in the iX App.
	Database *(only when **Advanced** is selected)*	Path to a database to be loaded automatically in the iX App. If the database or table is invalid or doesn't exist, an empty viewer is displayed.
	Table Name	Specifies the table of the selected database to be loaded in the iX App.
	Browse Data Table *(only when **Advanced** is selected)*	Opens a dialog to browse and select a database and its table. Once selected, the path and table name fields are automatically filled in the **Database** and **Table Name** fields.

NOTE
Simulation of this object is supported on the development PC only for PC targets.

TIP
For detailed information on databases, see the Databases chapter.

10.1.2.11. Digital Clock Object

The **Digital Clock** displays date, time and/or day of week.

The **time zone** is selected in the Time Zone and Region Group and is transferred to the target device when the project is downloaded. If no time zone is chosen, the current time zone of the panel or PC will remain.

The **date and time format** (e.g., 12-hour AM/PM or 24-hour) is based on the operating system settings. For HMI panel projects, this can be adjusted by selecting a different region in the Time Zone and Region Group. For PC projects, the region settings are adjusted in the local system account.

To configure object settings, select the object and go to the General Ribbon Tab. The following settings are available:

Parameter		Description
Display Format	**Date and Time**	Displays the date, time, or both on the object.
	Show Day of Week	Displays the current weekday on the object
	Show Seconds	Displays seconds if time is displayed.

10.1.2.12. Linear Meter Object

The **Linear Meter** displays a numeric value connected to a tag, representing it with a graphical meter. The meter can be styled and oriented either horizontally or vertically.

To configure object settings, select the object and go to the General Ribbon Tab. The following settings are available:

Parameter		Description
Value Scale	**Min Value**	Start and end values of the scale. Values can include one decimal digit.
	Max Value	
	Major Ticks	Number of labeled scale marks.
	Minor Ticks	Number of ticks between two adjacent major ticks.
Style		Predefined graphical styles for the object. The smaller styles are intended to allow full visibility in the smaller HMI panels.
Tag/Security	**Select Tag**	Connects the object to a tag from the Tags Configuration Page.
	Select Security Groups	Restricts access based on security groups. If empty, access is unrestricted.
	Select Visibility	• **Default**: Inherits settings from the Security Properties Dialog. • **Disabled**: The object is visible but grayed out for unauthorized users. • **Hidden**: The object is invisible for unauthorized users. • **Normal**: The object appears normally but cannot be interacted with by unauthorized users.

Parameter		Description
Visibility	**Scale**	Activates scale color. Select scale color from **Home** > **Format** > **Other Colors**. When disabled, scale color is transparent
	Transparent	Makes the object's frame and background color invisible.

10.1.2.13. Multi Picture Object

The **Multi Picture Object** allows the use of a set of pictures as a single object. The pictures change in the iX App depending on the intervals of a tag value. Imported pictures are added to the **Project Pictures** folder in the Component Library Window.

To configure object settings, select the object and go to the General Ribbon Tab. The following settings are available:

Parameter		Description
Settings	**Configure Pictures**	Configures the start/end values of intervals and selects pictures for each interval. Pictures can be selected from the **Project Pictures** folder in the Component Library Window or from the hard disk. Transparency can also be set. The original picture size is displayed. Use the **Add/Delete** buttons to add and remove items. **NOTE** Avoid using semi-transparent images as HMI panels not fully support opacity.
	Reset Picture	Restores original height and width to the picture.
	Stretch	Adapts height and width to the selection frame when resizing the picture.
	Lock Aspect Ratio	Maintains the height/width ratio of the object.

10.1.2.14. Picture Object

The **Picture Object** allows you to use an image as a functional object. You can import images into the project by browsing your PC for graphic files. Imported images will be stored in the **Project Pictures** folder in the Component Library Window.

The displayed picture can also be dynamic; depending of a set of pictures that have been linked to intervals in a tag value.

To configure object settings, select the object and go to the General Ribbon Tab. The following settings are available:

Parameter		Description
Settings	**Change Picture**	Opens a browse dialog to change picture. **NOTE** Avoid using semi-transparent images as HMI panels not fully support opacity.
	Reset Picture	Restores original height and width to the picture.
	Stretch	Adapts height and width to the selection frame when resizing the picture.
	Transparent	Makes the object's frame and background color invisible.

10.1.2.15. Roller Panel Object

The **Roller Panel Object** displays predefined texts for certain tag values.

To configure object settings, select the object and go to the General Ribbon Tab. The following settings are available:

Parameter		Description
Settings	**Title**	Sets the title for the roller panel.
	Show Title	Displays the title on the roller panel.
	Loop Scroll	Enables continuous scrolling through values.
	Disable Operator Input	Prevents users from editing the roller panel.
Text	**Configure Texts**	Click to open the Configure Texts Window and link the text to a tag.
Touch Settings	**Scroll Sensitivity**	Adjusts the sensitivity of the scrolling.
	Scroll Friction	Adjusts the friction during scrolling.

NOTE
Avoid using braces **{}** around digits in text objects.

TIP
You can also set dynamics (tag dependent values) for the roller panel properties in the Dynamics Ribbon Tab.

Configure Texts Window

The **Configure Texts** window allows you to set the following properties:

Parameter	Description
Select tag	Connects the texts to a specific tag.
Add	Adds a new text entry to the list.
Delete	Removes a text entry from the list.
Text	The text that will be displayed. **NOTE** Avoid using braces **{}** around digits in text objects.
Start Value	The minimum and maximum values at which the text is displayed. If the tag value falls outside the defined range, the text field value is displayed.
End Value	

10.1.2.16. Slider Object

The **Slider Object** is used to adjust a numeric value. When the slider is moved, it sets a value to a connected tag. If the tag value changes for any reason, the slider handle will display the current value in the iX App.

To configure object settings, select the object and go to the General Ribbon Tab. The following settings are available:

Parameter		Description
Value Scale	**Min Value**	Start and end values of the scale. Values can include one decimal digit.
	Max Value	
	Major Ticks	Number of labeled scale marks.
	Minor Ticks	Number of ticks between two adjacent major ticks.
Style		Predefined graphical styles for the object. The smaller styles are intended to allow full visibility in the smaller HMI panels.
Tag/Security	**Select Tag**	Connects the object to a tag from the Tags Configuration Page.
	Select Security Groups	Restricts access based on security groups. If empty, access is unrestricted.
	Select Visibility	• **Default**: Inherits settings from the Security Properties Dialog. • **Disabled**: The object is visible but grayed out for unauthorized users. • **Hidden**: The object is invisible for unauthorized users. • **Normal**: The object appears normally but cannot be interacted with by unauthorized users.
Visibility	**Scale**	Activates scale color. Select scale color from **Home** > **Format** > **Other Colors**. When disabled, scale color is transparent
	Transparent	Makes the object's frame and background color invisible.

10.1.2.17. Text Object

The **Text Object** displays read-only information for the operator. The displayed text can be static, or linked to intervals in a tag value that contain different strings.

To set the text to depend directly on a tag value, go to **Dynamics** > **General** > **General** .

To configure object settings, select the object and go to the General Ribbon Tab. The following settings are available:

Parameter		Description
Text	**Text**	Sets the text displayed on the object. **NOTE** Avoid using braces **{}** around digits in text objects.
	Multiline	Allows text to run over multiple lines using **Enter**. Confirm the text entry with **Ctrl** + **Enter**.
	Word Wrap *(not available when **Auto Size** is selected)*	Wraps the text to fit within the button's width.
	Configure Texts	Opens the Configure Texts Window to link the text to a tag.
	Auto Stretch Font *(not available when **Auto Size** is selected)*	Resizes the font to fit the height of the object.
	Auto Size *(default)*	Resizes the object according to the size of the default text string length and font size.
Text Alignment		Adjusts the text horizontally or vertically.

Configure Texts Window

The **Configure Texts** window allows you to set the following properties:

Parameter	Description
Select tag	Connects the texts to a specific tag.
Add	Adds a new text entry to the list.
Delete	Removes a text entry from the list.
Text	The text that will be displayed. **NOTE** Avoid using braces **{}** around digits in text objects.
Start Value	The minimum and maximum values at which the text is displayed. If the tag value falls outside the defined range, the text field value is displayed.
End Value	

10.1.2.18. Touch Combo Box Object

The **Touch Combo Box Object** allows users to create text selections from a drop-down list.

To configure object settings, select the object and go to the General Ribbon Tab. The following settings are available:

Parameter		Description
Settings	**Auto Size Item Height** *(default)*	Enables auto size of the item height.
	Item Height *(not available when **Auto Size Item Height** is selected)*	Sets the height of the individual items.
	ArrowBox Width	Sets the width of the arrow box.
	ScrollBar Width	Sets the width of the scroll bar.
Text	**Configure Texts**	Opens the Configure Texts Window to link the text to a tag.
Touch Settings	**Scroll Sensitivity**	Adjusts the sensitivity of the scrolling.

TIP
To set dynamics, i.e. tag dependent values for the object properties, go to Dynamics Ribbon Tab.

Configure Texts Window

The **Configure Texts** window allows you to set the following properties:

Parameter	Description
Select tag	Connects the texts to a specific tag.
Add	Adds a new text entry to the list.
Delete	Removes a text entry from the list.
Text	The text that will be displayed.  **NOTE** Avoid using braces **{}** around digits in text objects.
Start Value	The minimum and maximum values at which the text is displayed. If the tag value falls outside the defined range, the text field value is displayed.
End Value	

10.1.2.19. Touch List Box Object

The **Touch List Box Object** manages a list of predefined texts.

To configure object settings, select the object and go to the General Ribbon Tab. The following settings are available:

Parameter		Description
Settings	**Separator Visible**	Determines whether the separator line between items is visible.
	Item Height	Sets the height of the individual items.
	ScrollBar Width	Sets the width of the scroll bar.
Text	**Configure Texts**	Opens the Configure Texts Window to link the text to a tag.
Touch Settings	**Scroll Sensitivity**	Adjusts the sensitivity of the scrolling.

TIP
To set dynamics, i.e. tag dependent values for the object properties, go to Dynamics Ribbon Tab.

Configure Texts Window

The **Configure Texts** window allows you to set the following properties:

Parameter	Description
Select tag	Connects the texts to a specific tag.
Add	Adds a new text entry to the list.
Delete	Removes a text entry from the list.
Text	The text that will be displayed.  **NOTE** Avoid using braces **{}** around digits in text objects.
Start Value	The minimum and maximum values at which the text is displayed. If the tag value falls outside the defined range, the text field value is displayed.
End Value	

10.1.2.20. Trend Viewer Object

The **Trend Viewer Object** displays the values of connected tags over time, allowing for a comprehensive view of data changes with multiple trend curves.

Trend viewer data sources

- **Log Item**: Connects to a data logger, storing tag values in the project database. For more information, see Data Logger Function and Databases.
- **Tag**: Displays real-time values stored in RAM cache. For more information, see Tags.

Data behavior

- When the system time is changed to a later time, data is logged with the new time, and the trend viewer displays a straight line between the last logged point of the old time and the first logged point of the new time.
- A gap will appear in the trend viewer curve if there is a power interruption or communication error.
- Multiple independent trend viewers can be defined on the same or different screens, but the number of curves is limited by the available HMI panel memory.
- A large number of trend viewer pens and short sampling intervals may affect communication performance.

To configure object settings, select the object and go to the General Ribbon Tab. The following settings are available:

Parameter		Description
Curves	**Edit Curves**	Opens the Configure Curves Window, where you can edit the trend viewer curves.
Dynamics	**Trend Dynamics**	Opens the Trend Dynamics Properties, where you can set tag-dependent values for trend viewer properties. Dynamic properties override the **Value Scale** settings.
Value Scale	**Min Value**	Sets the start and end value of the scale.
	Max Value	
	Major Ticks	Sets the number of labeled scale marks on the scale.
	Minor Ticks	Sets the number of scale ticks between two adjacent major ticks.
Time Scale	**Time Range**	Sets the visible time span for the trend viewer in **Hour:Minute** format (00:00:00 to 23:59:59).
	Major Ticks	Sets the number of labeled time marks. This property is ignored in Historical Mode in Trend Viewer.
Style		Sets predefined graphical styles for the trend viewer.
Visibility	**Scale**	Activates scale color. Select scale color from **Home** > **Format** > **Other Colors**. When disabled, the scale color is transparent.
	Grid	Activates grid color. Select grid color from **Home** > **Format** > **Other Colors**. When disabled, the grid color is transparent.

Configure Curves Window

The **Configure Curves** dialog allows you to manage and customize the curves displayed in the Trend Viewer Object.

To open the **Configure Curves** window:

1. Select the Trend Viewer Object.
2. Click **General** > **Curves** > **Edit Curves**.

Settings

Parameter	Description
Add	Adds a new curve to the trend viewer object.

Parameter	Description
Remove	Removes the selected curve from the trend viewer object.
Name	Sets the symbolic name of the curve. Visible if a Trend Viewer Legend is used.
Tag	Sets the tag used to present values on the curve.
Log Item	Sets the data item designated for logging values.
Expression	Evaluates and applies an expression when drawing the curve. Expressions can be used on the log item or tag. For more information, see chapter Expressions.
Color	Sets the curve's color.
Thickness	Sets the line thickness of the curve.
Minimum Value Tag	Dynamically sets the Y-axis minimum value, allowing the curve to scale (optional).
Maximum Value Tag	Dynamically sets the Y-axis maximum value, allowing the curve to scale (optional).

TIP

Due to known issues, adding multiple curves connected to the same tag or log item is not permitted. If you attempt this, an error message will appear, and you won't be able to save until the issue is resolved.

Trend Dynamics Properties

Parameter		Description
Value Scale	**Min**	Sets the minimum/maximum value of the Y-scale. **Design Mode Value** is used for visualization when the tag values do not affect the minimum/maximum values.
	Max	
Time Scale	**Time Span**	Sets the time span of the X-scale in seconds. **NOTE** If changed, the trend viewer will fetch old data from the data logger for display. This process may take time, especially if the trend viewer has multiple curves or if multiple trend viewers use the same tag for changing the time span. It is advisable to limit the number of trend viewers controlled by the same tag for time span adjustments. Trend viewers that log real-time values and are not connected to a data logger will not reload values when the time span changes.
	Time Offset in History Mode	Enables using an offset to step back in history mode. The start value is when history mode was entered. The unit is in seconds.

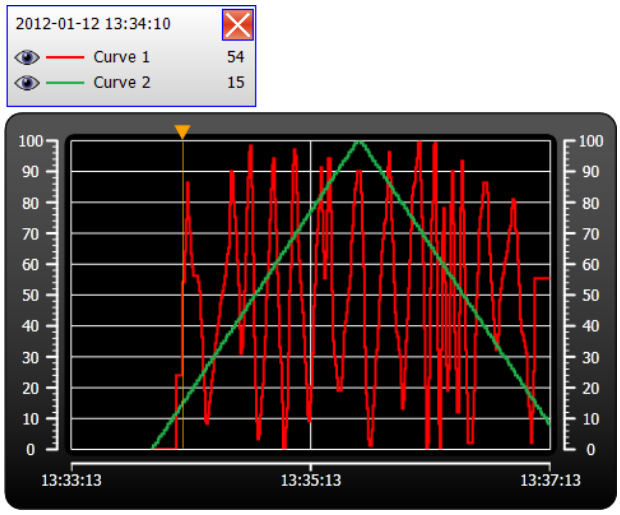
How to Use the Trend Viewer Object

Trend Viewer Legend

The **Trend Viewer Legend** is a window that displays the name and color of the curves in the current trend viewer object. In the iX App, you can display the trend viewer legend using the **Show Trend Viewer Legend** action.

The legend shows the values of all curves at a specific point when the needle is used directly in the trend viewer.

You can toggle the visibility of curves by clicking the visibility icon (eye) next to each curve.



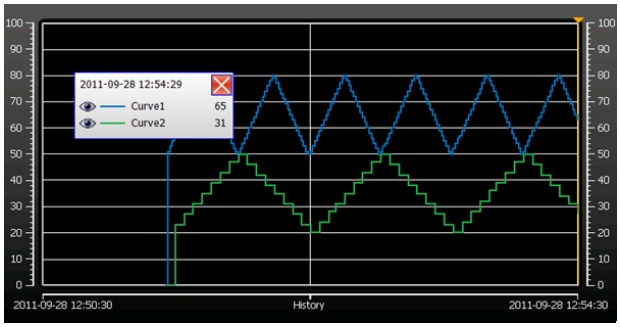
Trend Viewer Historical Mode

To display historical data in a trend viewer, connect the curves to log items from a data logger (see the Data Logger Function chapter). The start and end points of the historical data will be shown on the time axis, with the word **History** visible in the center.

Trend actions like button-click events must be defined to show or close the historical trend viewer.

Users can be allowed to select a specific time period to view the historical data.

The settings in **General** > **Time Scale** > **Major Ticks** are ignored in historical mode.



Show Logged Data in the Trend Viewer

To show logged data in the Trend Viewer Object:

1. Select the Trend Viewer Object.
2. Open the Configure Curves Window by clicking **General** > **Curves** > **Edit Curves**.
3. Add a curve and select the logged data as **Log Item**..

IMPORTANT
When displaying log items in a trend viewer, be aware of the log period limits for real-time values. For HMI panels, this limit is once per second, while for PC projects, it is 100 milliseconds. If a trend displays values from a data logger that logs more frequently than these limits, the curves may appear different in history mode compared to real-time mode. History mode shows all logged values without interval restrictions, which can lead to discrepancies.

10.1.3. Media Controls Objects

Media Controls objects allow the integration and management of multimedia content within the HMI panel.

Media Controls objects

Symbol	Name
	Web Browser Object

10.1.3.1. Web Browser Object

The **Web Browser** object allows you to display web pages within the HMI panel. It can be configured for regular web browsing or restricted to specific web pages or local files, especially useful when the panel is not connected to the Internet. Local files should be placed in the Project Files folder and accessed with the format **file:///filename.html** , for example **file:///ReadMe.html**.

Prerequisites

- **Microsoft WebView2 Runtime** is required for **simulation**.

To configure object settings, select the object and go to the General Ribbon Tab. The following settings are available:

Parameter		Description
Settings	**Address**	Sets the default web page.
	Home	Sets the web page displayed when clicking the **Home** button.
	Favorites	Opens the **Favorites** dialog, where you can set the web pages in the **Favorites** drop-down in the iX App.
	Address Read Only	Disables text entry in the address field.
	Address Field	Makes the address field visible.
	Go Button	Makes the **Go** button visible.
	Home Button	Makes the **Home** button visible.
	Back Button	Makes the navigation buttons visible.
	Forward Button	

Additional Settings for the Web Browser Object

The Property Grid Window has additional settings for the Web Browser Object. To access these properties, right-click on the **Web Browser** object and select **Properties**. In the **Extended** group, the following settings are available:

Parameter	Description
Block download of files	Prevents the user from downloading files.
Pop-up links	Blocks pop-up windows when clicking a link, or navigates to the link in the browser window.

How to add Videos and PDF Files to the Web Browser Object

VIDEO TUTORIAL

https://player.vimeo.com/video/1072168144?badge=0&autopause=0&player_id=0&app_id=58479

10.1.4. Tools for Debug Objects

Tools for Debug objects provide resources for troubleshooting and monitoring.

Tools for Debug objects

Symbol	Name
	Alarm Distributor Viewer Object
	Tag Monitor Object

10.1.4.1. Alarm Distributor Viewer Object

The **Alarm Distributor Viewer** object is a debug tool used for testing and verifying the Alarm Distributor Function in the iX App.

Some configuration is also possible within the iX App. On the **Alarm Distributor Viewer**, click **Settings** to edit the address book and configure alarm distribution settings.

10.1.4.2. Tag Monitor Object

The **Tag Monitor Object** allows you to display and modify tag values in the system, which is useful for system diagnostics.

To configure object settings, select the object and go to the General Ribbon Tab. The following settings are available:

Parameter		Description
Buttons	**Position**	Sets the position for the buttons.
	Delete Selected	Deletes the selected tag.
	Add	Adds a tag.
	Clear	Deletes all tags.
Display Settings	**Configure Columns**	Opens a dialog where you can configure the columns in the tag monitor object.
	Show Column headers	Enables column headers in the tag monitor object.

10.1.5. Windows Controls Objects

Windows Controls objects are interactive elements that can be added to a project to create functional and dynamic windows. These controls enable users to build responsive displays with components like buttons, sliders, and text fields. The following objects are available:

Windows Controls objects

Symbol	Name
	Check Box Object
	Combo Box Object
	Group Box Object
	List Box Object
	Progress Bar Object
	Radio Button Object
	Text Box Object

10.1.5.1. Check Box Object

The **Check Box** controls a digital tag value. The connected tag receives the value 1 when the box is checked and 0 when it is unchecked.

Adjust text formatting in **Home** > **Font**.

NOTE
Changing font family is only supported for projects designed for iX TxC andPC targets.

To configure object settings, select the object and go to the General Ribbon Tab. The following settings are available:

Parameter		Description
Text	**Text**	Sets the text displayed on the object.
	Multi Line	Allows text to run over multiple lines using **Enter**. Confirm the text entry with **Ctrl** + **Enter**.
	Word Wrap *(not available when **Auto Size** is checked)*	Wraps the text to fit within the button's width.
	Auto Strech Font *(not available when **Auto Size** is checked)*	Resizes the font to fit the height of the object.
	Auto Size	Resizes the object according to the size of the default text string length and font size.
Text Alignment		Adjusts the text horizontally or vertically.

10.1.5.2. Combo Box Object

The **Combo Box** is used to create selections from a drop-down list. When a keyboard is connected to the HMI panel, selections can also be made by typing.

To configure object settings, select the object and go to the General Ribbon Tab. The following settings are available:

Parameter		Description
Text	**Configure Texts**	Opens the Configure Texts Window to link the text to a tag.

Configure Texts Window

The **Configure Texts** window allows you to set the following properties:

Parameter	Description
Select tag	Connects the texts to a specific tag.
Add	Adds a new text entry to the list.
Delete	Removes a text entry from the list.
Text	The text that will be displayed.  **NOTE** Avoid using braces **{}** around digits in text objects.
Start Value **End Value**	The minimum and maximum values at which the text is displayed. If the tag value falls outside the defined range, the text field value is displayed.

How to Use the Combo Box Object

Enter Text for the Combo Box Object

Instructions

1. When the object is selected, go to **General** > **Text** > **Configure Text**.
2. Select a tag.
3. Click **Add** until there are enough rows in the table.
4. Enter text and values.

You can also use script code to control the contents of a combo box, see example below:

Example: Enter text with script code

The following script code relies on the existence of a combo box (ComboBox1) and a set of predefined recipes (“Book” and “TV”).

```
void Screen1_Opened(System.Object sender, System.EventArgs e)
{
    ComboBox1.Items.Clear();
    ComboBox1.Items.Add("Book");
    ComboBox1.Items.Add("TV");
}

void ComboBox1_SelectionChanged(System.Object sender, System.EventArgs e)
{
    string selectedItem = ComboBox1.SelectedItem as string;
    Globals.Recipe1.LoadRecipe(selectedItem);
}
```

When the screen opens, the script clears the box and writes the recipe name strings in the box. Selection from the combo box triggers loading of a recipe.

TIP

- For information on scripting, refer to the Scripts section.
- For information on recipes, refer to the Recipe Function section.

10.1.5.3. Group Box Object

The **Group Box** object adds a labeled frame around a group of objects.

10.1.5.4. List Box Object

The **List Box** object handles a list of predefined texts.

To configure object settings, select the object and go to the General Ribbon Tab. The following settings are available:

Parameter		Description
Text	**Configure Texts**	Opens the Configure Texts Window to link text to a tag.

Configure Texts Window

The **Configure Texts** window allows you to set the following properties:

Parameter	Description
Select tag	Connects the texts to a specific tag.
Add	Adds a new text entry to the list.
Delete	Removes a text entry from the list.
Text	The text that will be displayed. **NOTE** Avoid using braces **{}** around digits in text objects.
Start Value **End Value**	The minimum and maximum values at which the text is displayed. If the tag value falls outside the defined range, the text field value is displayed.

How to Use the List Box Object

Enter Text for the List Box Object

Instructions

1. When the object is selected, go to **General** > **Text** > **Configure Text**.
2. Select a tag.
3. Click **Add** until there are enough rows in the table.
4. Enter text and values.

You can also use script code to control the contents of a combo box, see example below:

Example: Enter text with script code

The following script code includes two objects on Screen1, a button (Button1) and a list box (ListBox1).

```
public partial class Screen1
{
    int count = 0;

    void Button1_Click(System.Object sender, System.EventArgs e)
    {
        ListBox1.Items.Add("Count " + count.ToString());
        count++;
    }
}
```

When the button is clicked, the script is triggered and a string is written to the list box and the value of an internal counter (count) is incremented.

TIP
For detailed information about scripting, refer to the Scripts section.

10.1.5.5. Progress Bar Object

The **Progress Bar** displays a meter without numeric scaling, representing an analog value. The value can be connected to a tag.

Properties such as orientation, minimum and maximum values, and meter color (foreground property) are configured in the Property Grid Window. To open the property grid window, **right-click** on the object and select **Properties**.

10.1.5.6. Radio Button Object

The **Radio Button** object controls the value of a connected digital tag.

Adjust text formatting in **Home** > **Font**.

To configure object settings, select the object and go to the General Ribbon Tab. The following settings are available:

Parameter		Description
Text	**Text**	Sets the text displayed on the object.  **NOTE** Avoid using braces **{}** around digits in text objects.
	Multi Line	Allows text to run over multiple lines using **Enter**. Confirm the text entry with **Ctrl** + **Enter**.
	Word Wrap *(not available when **Auto Size** is checked)*	Wraps the text to fit within the button's width.
	Auto Strech Font *(not available when **Auto Size** is checked)*	Resizes the font to fit the height of the object.
	Auto Size	Resizes the object according to the size of the default text string length and font size.
Text Alignment		Adjusts the text horizontally or vertically.

How to Use the Radio Button Object

Group Radio Buttons for Tag Control

Utilize script code to enable a group of radio buttons to control a tag value collectively. Only one radio button can be active (set to 1) at any given time, with buttons grouped by their respective screen.

Example

The following example illustrates a setup with three radio buttons and a tag of type int16.

```
public partial class Screen1
{
    void Screen1_Opened(System.Object sender, System.EventArgs e)
    {
        Globals.Tags.Tag1.ValueChange += Tag1_ValueChanged;
        SetRadioButtonState(Globals.Tags.Tag1.Value);
    }

    void Screen1_Closed(System.Object sender, System.EventArgs e)
    {
        Globals.Tags.Tag1.ValueChange -= Tag1_ValueChanged;
    }

    private void Tag1_ValueChanged(System.Object sender, System.EventArgs e)
    {
        SetRadioButtonState(Globals.Tags.Tag1.Value);
    }

    private void SetRadioButtonState(int Value)
    {
        RadioButton1.Checked = (Value == 1);
        RadioButton2.Checked = (Value == 2);
        RadioButton3.Checked = (Value == 3);
    }

    void RadioButton1_Click(System.Object sender, System.EventArgs e)
    {
        Globals.Tags.Tag1.Value = 1;
    }

    void RadioButton2_Click(System.Object sender, System.EventArgs e)
    {
        Globals.Tags.Tag1.Value = 2;
    }

    void RadioButton3_Click(System.Object sender, System.EventArgs e)
    {
        Globals.Tags.Tag1.Value = 3;
    }
}
```

TIP
For detailed information about scripting, refer to the Scripts section.

10.1.5.7. Text Box Object

The **Text Box** object is a multi-line text box that allows users to input text in the iX App. By default, the text is editable, but it is not saved; it resets when the screen changes. The text box can also be configured as read-only.

To configure object settings, select the object and go to the General Ribbon Tab. The following settings are available:

Parameter		Description
Text	**Text**	Sets the text displayed on the object.
	Multi Line	Allows text to run over multiple lines using **Enter**. Confirm the text entry with **Ctrl** + **Enter**.
	Word Wrap	Wraps the text to fit within the button's width.
	Read Only	Disables input from the iX App.
Text Alignment		Adjusts the text horizontally or vertically.
Visibility	**Transparent**	Makes the object's frame and background color invisible.

10.1.6. Additional Controls Objects

iX Developer lets you use .NET controls included in the .NET Framework. These objects appear as **Additional Controls** in the Objects Group in the Home Ribbon Tab.

CAUTION
You can use these objects directly in iX Developer; however, they are not officially supported by Beijer Electronics. You are responsible for how the objects are used and how they function in your application.

TIP
In iX Developer 3, **NuGet** replaces the previous **Reference Assemblies** setup. For detailed information, see the NuGet Packages section.

10.1.6.1. Add Additional Controls to the Objects Group

To add Additional Controls Objects to the Objects Group in the Home Ribbon Tab:

1. Go to **Home** > **Objects**. Expand the objects group by clicking the arrow in the lower-right corner.
2. Click **Add Control**.

> **NOTE**
> The **Screen Carousel Object** is removed starting from iX Developer version 2.51.

3. Choose from the available **Default Controls**.

> **NOTE**
> Additional controls can be used "as is," but they are not officially supported by Beijer Electronics. Users of iX Developer are responsible for ensuring the functionality of any added controls.

4. Click **OK**.

The added controls will now appear under **Additional Controls** in the expanded objects group.

> **NOTE**
> DLL-based support for additional controls was removed in iX Developer version 3.0. To add custom controls, use NuGet Packages.

10.2. How to Work with Objects

For basic information on how to work with objects in iX Developer, refer to Get Started with Objects.

> **TIP**
> For instructions on how to optimize object usage, see Optimize Object Usage.

10.2.1. Create Objects

You can create objects using the following methods:

- **Click**: To insert a default-sized object, click the object, then click the position on the screen where you want to place it.
- **Click and drag**: To insert an object of your desired size, click the object, then click and drag to the desired size at the desired position on the screen.

> **TIP**
> To maintain the object's aspect ratio while creating it with **Click and drag**, hold **Shift**.

10.2.1.1. Create Series of Objects

The **Create Series** function enables the creation of multiple identical objects, including graphic elements and controller connections. Follow these steps to create a series of objects:

1. Create an object.
2. Right-click on the object and select **Create series**.
3. Specify the parameters.

Parameter	Description
Number of copies	Enter the desired number of columns and rows to create.
Spacing	Define the pixel spacing between the added objects.
Address incrementation	Choose how addresses for the added objects will be incremented. Only addresses available in the driver will be included.
Increment direction	Specify whether addresses should be incremented horizontally or vertically.

4. Click **OK**.

Graphical objects will be displayed on the screen, and corresponding tags will be added to the project.

TIP
To create series of tags without adding graphic objects, go to **Project Explorer** > **Functions** > **Tags**, right-click on the tag and select **Create series**.

10.2.2. Modify Objects

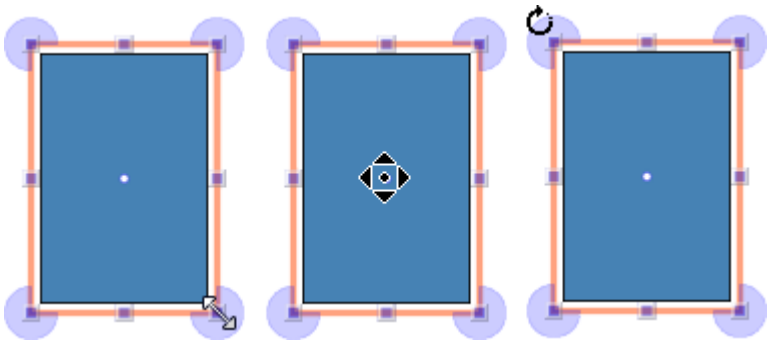
10.2.2.1. The Select Tool

The **Select tool** is the default tool when editing screens and is located at the top left of the Objects Group when the group is not expanded. After using any other object control, the mouse pointer will revert to the **Select tool**.

The **Select tool** allows you to:

- **Select** objects
- **Move** objects
- **Resize** objects
- **Rotate** objects

The appearance of the tool changes when hovering over an object, indicating different functions such as resizing or rotating, see picture below.



The select tool appearance for Resize, Rotation center, and Rotate

Select Multiple Objects

You can select multiple objects using the following methods:

- **Drag Selection**: Place the cursor outside the objects and draw a frame around them.
- **Ctrl Selection**: Select one object, then hold **Ctrl** and click on additional objects.

In a multiple selection, the primary object is indicated with a **red frame**, while other selected objects are marked with **blue frames**. Clicking on any object in the selection will make it the primary selection.

A multiple selection can be used to arrange objects via **Arrange** in the Format Group. The object with the primary selection is used as guide object.

The ribbon groups displayed in the General Ribbon Tab and Actions Ribbon Tab will reflect the properties common to the selected objects. Changes made will affect all selected items.

10.2.2.2. Move Object

An object can be moved in the following ways:

- **Drag**: Select and drag the object using the The Select Tool.
- **Arrow Keys**: Select the object with the The Select Tool, then use the arrow keys on your keyboard to move the object.
- **Property Grid**: Open the Property Grid Window and edit the **Left** and **Top** coordinates to set the desired position.

Copy and Move Object

To make a copy of an object and move it:

- Hold **Ctrl** while selecting and dragging the object with the The Select Tool.

10.2.2.3. Resize Object

The size of an object is part of the properties for the object and can be changed in the following ways:

- **Drag**: Use the The Select Tool to resize the object by dragging its corners or edges.
- **Arrow keys**: To resize the selected object one pixel at the time, select the object with the The Select Tool, hold **Shift** and use the arrow keys on the keyboard.
- **Property grid**: Open the Property Grid Window and edit the **Height** and **Width** values.

10.2.2.4. Rotate Object

NOTE
Rotate object is currently not supported.

10.2.2.5. Hide or Lock Objects

You can hide or lock objects using the following methods:

- **Object browser**: In the Object Browser Window, click the **eye** to toggle visibility or the **lock** icon to change the locked state.
- **Right-click**: Right-click on the object and choose **Hide Selection** or **Lock Selection**.
- **Property grid**: In the Property Grid Window, check or uncheck the **Visibility** option to control the object's visibility.

10.2.3. Arrange Objects

Use the **Arrange** control in **Home** > **Format** to order, group, resize, align, and distribute screen objects.

TIP
You can also use the Object Browser Window to reorder objects.

Command		Description
Order Objects	**Bring to Front**	Positions the selected object on top of any overlapping objects.
	Send to Back	Positions the selected object behind any overlapping objects.
	Bring Forward	Makes the selected object swap place with the nearest overlapping object.
	Send Backward	Makes the selected object swap place with the nearest obscured object.
Group Objects	**Group**	Groups selected objects, allowing them to be moved and resized as one. The primary selection (orange frame) can be changed by clicking another object. Grouped objects can also be saved in the Component Library Window for reuse.
	Ungroup	Separates grouped objects into individual objects. Any properties applied to the group will remain on applicable individual objects.
Make Same Size	**Make Same Width**	Sets all selected objects to the same width, using the primary selection as the reference.
	Make Same Height	Sets all selected objects to the same height, using the primary selection as the reference.
	Make Same Size	Sets all selected objects to the same width and height, using the primary selection as the reference.

Command		Description
Position Objects	**Align**	Aligns and distributes objects vertically and horizontally using the primary selection as guide object. Objects can be aligned by their center or their edges (top, bottom, right, left).

10.2.4. Copy and Apply Formatting

Follow these steps to copy and apply formatting from one object to another:

1. Select the object with the desired format and font properties.
2. Click **Home** > **Clipboard** > **Format Painter**. The mouse pointer changes to a paintbrush.
3. Click the object to apply the formatting. The format properties of the first object will be applied to the selected object.

NOTE

- Customizations made using the Other Colors control are not transferred when using the **Format Painter**.
- Formatting can be copied between objects of different types (e.g., from a rectangle to an HMI control).

10.2.5. Blinking Objects

Some objects can be displayed as blinking objects. Blinking is enabled and edited from the **Edit Blink Dynamics** dialog. The **Edit Blink Dynamics** dialog can be opened in two ways:

- In the Property Grid Window, click **Animations** > **Blink**.
- Click **Dynamics** > **General** > **Blink**.

The following blink dynamics settings are available:

Parameter	Description
Blink Enabled	Select tag to enable or disable blinking depending on the selected tag value. When the tag value is equal to the given value (=**True**), blink is enabled. When the tag value is not equal to the given value (=**False**), blink is disabled.
Blink Interval	• **Constant blink intervals**: Set the time in milliseconds (ms) for the object to be invisible before it reappears. The default is 1,200 ms, with a maximum of 10,000 ms. For HMI panel projects, the minimum is 400 ms, and it is recommended to use multiples of 400 ms for a harmonious blink experience. For PC projects, the minimum is 200 ms. • **Blink intervals bound to a tag**: Select a tag to change blink intervals based on that tag.

NOTE

- It is not recommended to use blinking for objects intended for user interaction.
- Security settings take precedence over blinking; an object hidden due to security cannot be made visible through blinking.

The following objects **do not** support blinking:

- Alarm Distributor Viewer
- Alarm Viewer
- Audit Trail Viewer
- Check Box
- Combo Box
- Database Viewer
- Group Box
- List Box
- Media Player
- PDF Viewer
- Progress Bar
- Radio Button
- Web Browser

10.2.6. Create and Apply Object Variations

The **Object Variations** feature allows you to apply preset styles to various objects. Default styles are available for objects such as **Button** and **Circular Meter**, while most other objects support user-defined styles.

Follow the steps below to add user-defined styles in the Object Variations Group:

1. Select an object as a starting point.
2. Modify the object using tools in the Home Ribbon Tab or Property Grid.
3. Right-click the modified object and select **Save Variations** from the context menu.

The new variation will be available in the Object Variations Group for the same object type.

Objects supporting the Save Variation command

- Rectangle
- Ellipse
- Analog Numeric
- Button
- Text
- Linear Meter
- Slider
- Circular Meter
- Trend Viewer
- Chart
- Action Menu
- Animated Label
- Digital Clock
- Alarm Viewer

The following **properties** are saved in object variations:

- Colors
- Size
- Font
- Rotation
- Visibility
- Security groups required
- Visibility on Access denied
- Delay Mouse / Touch Input
- Horizontal and vertical alignment
- picture horizontal and vertical alignment
- Audit trail description.

The following **object specific properties** are also saved:

Object	Properties
Action Menu Object	View type, Orientation
Alarm Viewer Object	Button position
Analog Numeric Object	Limit number of characters, Number of characters, Number of decimals, Prefix, Suffix
Animated Label Object	Animation direction, View type
Chart Object	Chart type 3D view, Show header, Show legend, Show scale
Digital Clock Object	Display format, Show seconds
Linear Meter Object	Orientation, Scale Transparency
Slider Object	Orientation, Scale Transparency
Text Object	Horizontal alignment, Vertical alignment
Trend Viewer Object	Major value ticks, Minor value ticks
Rectangle	Radius, Effects

NOTE
Features only supported in PC environment may not work on other platforms.

10.2.7. Object Access and Visibility

You can limit object access and visibility based on user groups.

TIP
For more information on users and user groups, refer to the User Management chapter.

Configure the object access and visibility:

1. Select the object.

2. Go to **General** > **Tag/Security**.
3. Select the required security group(s) and choose a visibility option.
   Visibility options
   - **Default**: Inherits settings from the Security Properties Dialog.
   - **Disabled**: The object is visible but grayed out for unauthorized users.
   - **Hidden**: The object is invisible for unauthorized users.
   - **Normal**: The object appears normally but cannot be interacted with by unauthorized users.

NOTE

- **Security overrides dynamics**: Security settings override dynamics.
- **Function keys**: Security settings cannot be applied to Function Keys.
- **Recommendation**: Use the **Hidden** option for sensitive functions such as service tools.

10.2.8. Dynamic Controls

Dynamic controls are used for changing the properties of an object depending on a changed tag value. The dynamic controls are found in the Dynamics Ribbon Tab.

To add a dynamic control to an object

1. Select the desired object.
2. Click the desired control in the Dynamics Ribbon Tab, configure the settings. and click **OK**.

A lock icon resembling a chain indicates that a dynamic setting has been applied to the selected object.

To remove a dynamic control from an object

1. Select the object with the dynamic control.
2. Click the dynamic control in the Dynamics Ribbon Tab, and click **Clear Dynamics**.

10.2.8.1. Priority of Settings

- **Dynamic** settings override settings from the Format Group in the Home Ribbon Tab.
- **Security** settings override **dynamic** settings.
- **Visibility** settings override **blinking** effects.
- If a **script** is applied to a property with dynamic settings, the dynamic settings are disabled.

10.2.8.2. Move Object Based on Tag Value Changes

Position coordinates on the screen are given in pixels of the screen resolution. The position 0,0 corresponds to the upper left corner of the screen.

To move an object vertically when a tag value changes, follow these steps:

1. Draw and position the object at the desired start position and keep it selected.
2. Click **Dynamics** > **Layout** > **Move**. This opens the **Edit Move Dynamics** dialog.
3. Select the tag that will control the movement.

4. Enter the **Tag Start Value** and **Tag End Value**.
   - Start and end positions are automatically suggested based on the current start position.
   - A ghost object will appear at the suggested finish position, with a guide line connecting it to the original object.
5. Move the ghost object to its final position on the screen.
   - The object's coordinates will be updated.
   - An animation of the movement will be shown. You can also manually adjust the coordinates by updating the **End Top** and **End Left** fields.
   - The start and end coordinates will limit the object's movement, even if the tag value exceeds the start or end value.
6. Click **OK**.

> **NOTE**
> When move or size dynamics is applied, moving or resizing of the object will be disabled. A lock icon resembling a chain in the upper left corner of the object (when selected) indicates that the object has a dynamic control applied and that it cannot be moved or resized.

10.2.8.3. Resize Object Based on Tag Value Changes

To dynamically resize an object when a tag value changes, follow these steps:

> **NOTE**
> Dynamic resizing of external picture files may result in a resolution loss if the picture is enlarged to a size larger than what is used elsewhere in the current project.

1. Draw and size the object at its initial size, then keep it selected.
2. Click **Dynamics** > **Layout** > **Size**. This opens the **Edit Size Dynamics** dialog.
3. Select the tag that will control the resizing.
4. Enter the **Start Size** and **Finish Size** values. These are measured in pixels of screen resolution.
   - The start size will be suggested based on the current size of the object.
   - A suggested finish size will be provided, showing an increase in both width and height.
   - A ghost object with the finish size will appear for reference.
   - Objects can only be resized from left to right and top to bottom.
5. Adjust the ghost object to its final size.
   The coordinates for the object is updated. An animation of the resizing appears on the screen. Coordinate values can also be updated directly in the numerical fields **End Width** and **End Height**. The coordinates for start and finish will limit the size of the object irrespective if the tag value is outside its start and end value.
   - The object's coordinates will update automatically, and an animation of the resizing will appear.
   - You can also update the **End Width** and **End Height** fields directly with numerical values.

- The start and finish coordinates will limit the object's resizing, even if the tag value exceeds the set range.

6. Click **OK**.

NOTE
When move or size dynamics is applied, moving or resizing of the object will be disabled. A lock icon resembling a chain in the upper left corner of the object (when selected) indicates that the object has a dynamic control applied and that it cannot be moved or resized.

10.2.8.4. Change Object Fill Color Based on Tag Value

To change the fill color of an object when a tag value changes, follow these steps:

1. Draw the object and keep it selected.
2. Click **Dynamics** > **Color** > **Fill**. This opens the **Edit Fill Color Dynamics** dialog.
3. Select the tag that will control the color changfe.
4. Enter the values that will trigger the color change.
5. Choose the desired color and gradient from the color menu.
6. Click **OK**.

10.2.8.5. Hide Object when Tag Value Changes

To hide an object when a tag value changes, follow these steps:

1. Draw the object and keep it selected.
2. Click **Dynamics** > **General** > **Visibility**.
3. Select the tag that will control the object's visibility.
   - The object will be visible when the tag reaches a specified value (treated as **True**).
   - All other values will make the object invisible (treated as **False**).

10.2.9. Connect Object to Text Library Texts

To connect the text on an object to a text library text, follow these steps:

1. Select the desired text object on the screen.
2. Click **Dynamics** > **General** > **General** to open the Edit General Dynamics Dialog.
3. Navigate to **Text**.
4. Select **Text Library Group Converter** in the first dropdown.
5. Select tag in the **Select Tag** dropdown.
6. Click **...** next to the **Select Tag** dropdown.
7. Define the **Expression Code**. For more information, see the Expressions chapter.
8. Click **OK**.

The object will now present the texts from the text library instead of the value.

10.2.10. Components

The Component Library Window contains a variety of predefined graphical objects organized into categories. You can also save user-defined graphical objects and other files within the component library. Note that all components in a subfolder must have unique names.

Plain text can be saved as a component, allowing you to create reusable script components by dragging a section of script code from the Script View Mode into the Component Library Window. Each component is displayed with its name and a thumbnail representation. Grouped objects are shown as a thumbnail representing the primary selection of the group, while text files and folders are represented by their operating system thumbnails.

NOTE
Only static graphical objects are supported in iX Developer.

The following chapters describe how to work with components.

10.2.10.1. Add Components to the Project

To add components to the project, you can either:

- Drag-and-drop the component from the Component Library Window to the current screen, or
- Use standard copy and paste commands.

10.2.10.2. Add Components to the Component Library Window

To add components (objects) to the Component Library Window, follow these steps:

- Copy the component from the current screen to the Component Library Window, using standard copy and paste commands.
  - A text file saved in the component library will be converted into a text box object when pasted into a screen.
  - A component copied from an object with a tag connection will retain its tag connection and any dynamic properties when used from the component library.
  - If a component with a tag association is reused in another project, ensure that all referenced tags are present in the new project.

10.2.10.3. Add a Category to the Component Library Window

To add a new category to the Component Library Window:

1. Right-click on an empty area of the component library window.
2. Select **Add Components** > **Add Category**.
3. Enter your preferred name for the new category.

NOTE
User-defined categories are not project-specific. This means that objects saved in a user-defined category will be accessible in all iX Developer projects.

10.2.10.4. Import and Export Components

Import and Export Components Between Projects

You cannot import or export single components between projects; however, entire component categories, along with their included components, can be imported or exported. To do this:

1. In the Component Library Window, right-click on the category folder.
2. Select **Import Library** or **Export Library**.
3. Choose the project file location for the import or export.

NOTE
The program does not indicate the location of the component files.

Import External Components into iX Developer

To import vector graphics as XAML components into iX Developer, see the following chapters.

XAML Guidelines

All information, including images, must be contained within the XAML file. Links to external objects will result in missing components. Additionally, the appearance of the XAML file must not depend on external inputs, such as parameters controlling multiple images based on internal logic within the XAML file.

XAML Naming Controls

Not supported:

```
<ContentControl
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    Width="120"
    Height="15"
    x:Name="Root">
    <TextBlock x:Name="Description" Text="This is a description" />
</ContentControl>
```

Supported:

```
<ContentControl
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    Width="120"
    Height="15">
    <TextBlock Text="This is a description" />
</ContentControl>
```

Width and Height

The root control must specify both `Width` and `Height` attributes.

Not supported:

```
<Rectangle
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:signatures="http://www.beijerelectronics.com/hmi/ix/signatures"
    Fill="Red" />
```

Supported:

```
<Rectangle
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:signatures="http://www.beijerelectronics.com/hmi/ix/signatures"
    Width="50"
    Height="50"
    Fill="Red" />
```

Bindings

The following properties are supported for XAML binding within components:

- `ISymbolViewboxSignature.Fill` - displayed as **Fill** in the Property Grid Window.
- `ISymbolViewboxSignature.Stroke` - displayed as **Outline** in the Property Grid Window.
- `ISymbolViewboxSignature.StrokeThickness` - displayed as **Outline Thickness** in the Property Grid Window.

Supported:

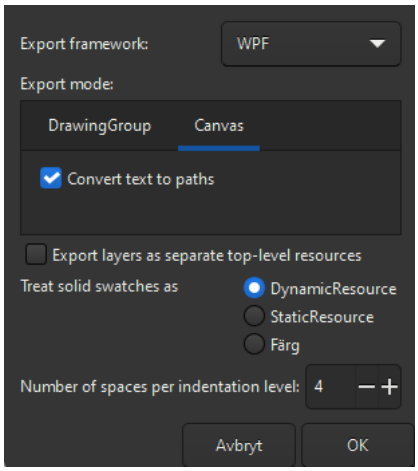
```
<Rectangle
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:signatures="http://www.beijerelectronics.com/hmi/ix/signatures"
    Width="50"
    Height="50"
    Fill="{Binding Fill, RelativeSource={RelativeSource ▶
AncestorType={x:Type signatures:ISymbolViewboxSignature}}}"
    Stroke="{Binding Stroke, RelativeSource={RelativeSource ▶
AncestorType={x:Type signatures:ISymbolViewboxSignature}}}"
    StrokeThickness="{Binding StrokeThickness, ▶
RelativeSource={RelativeSource AncestorType={x:Type ▶
signatures:ISymbolViewboxSignature}}}" />
```

NOTE
Binding to properties outside of the component is not supported.

Basic XAML Import

1. Install **Inkscape**. Download the free software from the Inkscape website.
2. Open **Inkscape** and draw your image.

3. Click **File** > **Save As**, select the **Microsoft XAML** format, and save the file using the recommended settings shown in the image.



4. Open the saved file with a text editor, for example **Notepad**.
5. Delete everything outside of the `<Canvas>` brackets, keeping only the first row: `<?xml version="1.0" ?>`.
6. Replace `UseLayoutRounding="False"` in the first `<Canvas>` line with:

```
xmlns:signatures="http://www.beijerelectronics.com/hmi/ix/signatures"
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
```

7. Remove any further occurances of `UseLayoutRounding="False"`.
8. Save the edited XAML-file.
9. Open iX Developer.
10. Right-click on a node in the **ComponentLibrary** in the Component Library Window and choose **Add Components**.
11. Select your XAML-file.
12. After importing, ensure to re-open the folder in the **Component Library** where you imported the file; otherwise, it might not display.

Example

Original Inkscape XAML File (not compatible with iX Developer):

```
<?xml version="1.0" ?>
<Grid xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml" ▶
xmlns:d="http://schemas.microsoft.com/expression/blend/2008" xmlns="http://
schemas.microsoft.com/winfx/2006/xaml/presentation">
    <Grid.Resources>
        <ResourceDictionary/>
    </Grid.Resources>
    <Viewbox Stretch="Uniform" Name="Lager_1">
        <Canvas UseLayoutRounding="False" Width="147.95735" ▶
Height="149.04263">
            <Canvas.Clip>
                <RectangleGeometry Rect="0.0,0.0,147.95735,149.04263"/>
            </Canvas.Clip>
            <Canvas UseLayoutRounding="False" ▶
```

```
RenderTransform="1.0,0.0,0.0,1.0,-27.131548,-9.0438461">
                <Rectangle RadiusX="0.0" RadiusY="0.0" ▶
Canvas.Left="27.131548" Canvas.Top="19.17296" Width="108.16443" ▶
Height="96.950058" Fill="#ff000000"/>
                <Ellipse Canvas.Left="41.601711" Canvas.Top="37.62241" ▶
Width="90.438484" Height="67.647988" Fill="#ff000000"/>
                <Path Fill="#ff000000">
                    <Path.Data>
                        <PathGeometry Figures="M 118.655 96.0457 A 37.4415 ▶
62.0408 0 0 1 81.5116 158.085 A 37.4415 62.0408 0 0 1 43.777 97.0327 A ▶
37.4415 62.0408 0 0 1 80.3203 34.0225 A 37.4415 62.0408 0 0 1 118.636 ▶
94.0719 l -37.4226 1.97382 z"/>
                    </Path.Data>
                </Path>
                <Path Fill="#ff000000">
                    <Path.Data>
                        <PathGeometry Figures="m 175.089 70.3611 a 25.3228 ▶
18.2686 0 0 1 -25.1213 18.268 a 25.3228 18.2686 0 0 1 -25.521 -17.9774 a ▶
25.3228 18.2686 0 0 1 24.7153 -18.554 a 25.3228 18.2686 0 0 1 25.9142 ▶
17.6822 l -25.31 0.581212 z" FillRule="Nonzero"/>
                    </Path.Data>
                </Path>
                <Path Fill="#ff000000">
                    <Path.Data>
                        <PathGeometry Figures="m 167.492 43.4105 a 9.58648 ▶
25.6845 0 0 1 -9.51022 25.6837 a 9.58648 25.6845 0 0 1 -9.66153 -25.2751 a ▶
9.58648 25.6845 0 0 1 9.35651 -26.0858 a 9.58648 25.6845 0 0 1 9.81039 ▶
24.8601 l -9.58163 0.817149 z" FillRule="Nonzero"/>
                    </Path.Data>
                </Path>
                <Path Fill="#ff000000">
                    <Path.Data>
                        <PathGeometry Figures="M 107.079 46.6663 A 26.9507 ▶
37.6224 0 0 1 80.3429 84.2875 A 26.9507 37.6224 0 0 1 53.1812 47.2648 A ▶
26.9507 37.6224 0 0 1 79.4854 9.05456 A 26.9507 37.6224 0 0 1 107.066 ▶
45.4693 L 80.1285 46.6663 Z"/>
                    </Path.Data>
                </Path>
            </Canvas>
        </Canvas>
    </Viewbox>
</Grid>
```

Edited XAML File (compatible with iX Developer):

```
<?xml version="1.0" ?>

        <Canvas xmlns:signatures="http://www.beijerelectronics.com/hmi/ix/
signatures"  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/
presentation" xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml" ▶
Width="147.95735" Height="149.04263">
            <Canvas.Clip>
                <RectangleGeometry Rect="0.0,0.0,147.95735,149.04263"/>
            </Canvas.Clip>
            <Canvas RenderTransform="1.0,0.0,0.0,1.0,-27.131548,-9.0438461">
                <Rectangle RadiusX="0.0" RadiusY="0.0" ▶
```

```
Canvas.Left="27.131548" Canvas.Top="19.17296" Width="108.16443" ▶
Height="96.950058" Fill="#ff000000"/>
                <Ellipse Canvas.Left="41.601711" Canvas.Top="37.62241" ▶
Width="90.438484" Height="67.647988" Fill="#ff00ff00"/>
                <Path Fill="#ff00c5c6">
                    <Path.Data>
                        <PathGeometry Figures="M 118.655 96.0457 A 37.4415 ▶
62.0408 0 0 1 81.5116 158.085 A 37.4415 62.0408 0 0 1 43.777 97.0327 A ▶
37.4415 62.0408 0 0 1 80.3203 34.0225 A 37.4415 62.0408 0 0 1 118.636 ▶
94.0719 l -37.4226 1.97382 z"/>
                    </Path.Data>
                </Path>
                <Path Fill="#ff0000ff">
                    <Path.Data>
                        <PathGeometry Figures="m 175.089 70.3611 a 25.3228 ▶
18.2686 0 0 1 -25.1213 18.268 a 25.3228 18.2686 0 0 1 -25.521 -17.9774 a ▶
25.3228 18.2686 0 0 1 24.7153 -18.554 a 25.3228 18.2686 0 0 1 25.9142 ▶
17.6822 l -25.31 0.581212 z" FillRule="Nonzero"/>
                    </Path.Data>
                </Path>
                <Path Fill="#ffff0000">
                    <Path.Data>
                        <PathGeometry Figures="m 167.492 43.4105 a 9.58648 ▶
25.6845 0 0 1 -9.51022 25.6837 a 9.58648 25.6845 0 0 1 -9.66153 -25.2751 a ▶
9.58648 25.6845 0 0 1 9.35651 -26.0858 a 9.58648 25.6845 0 0 1 9.81039 ▶
24.8601 l -9.58163 0.817149 z" FillRule="Nonzero"/>
                    </Path.Data>
                </Path>
                <Path Fill="#ff005500">
                    <Path.Data>
                        <PathGeometry Figures="M 107.079 46.6663 A 26.9507 ▶
37.6224 0 0 1 80.3429 84.2875 A 26.9507 37.6224 0 0 1 53.1812 47.2648 A ▶
26.9507 37.6224 0 0 1 79.4854 9.05456 A 26.9507 37.6224 0 0 1 107.066 ▶
45.4693 L 80.1285 46.6663 Z"/>
                    </Path.Data>
                </Path>
            </Canvas>
        </Canvas>
```

Import XAML with Dynamic Properties

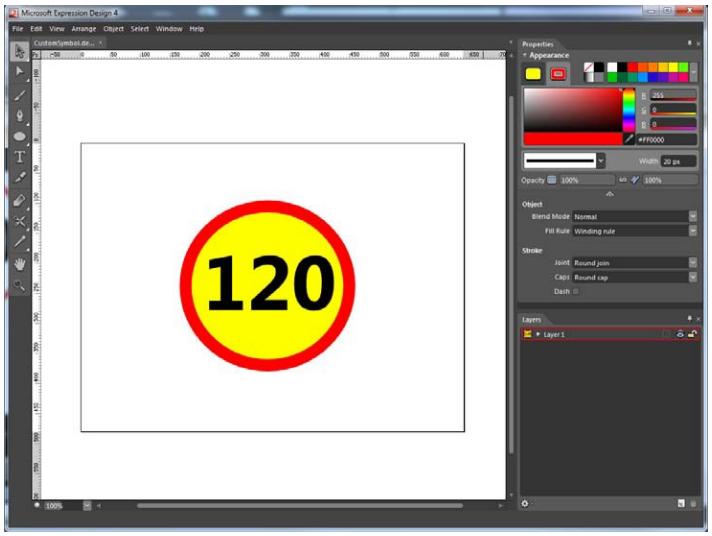
You can bind some component properties directly to tags in iX Developer's Property Grid Window to make the dynamic. The following properties are supported for binding:

```
//Namespace
xmlns:signatures="http://www.beijerelectronics.com/hmi/ix/signatures"

//Fill
Fill="{Binding Fill, RelativeSource={RelativeSource AncestorType={x:Type ▶
signatures:ISymbolViewboxSignature}}}"

//Stroke
Stroke="{Binding Stroke, RelativeSource={RelativeSource ▶
AncestorType={x:Type signatures:ISymbolViewboxSignature}}}"
```

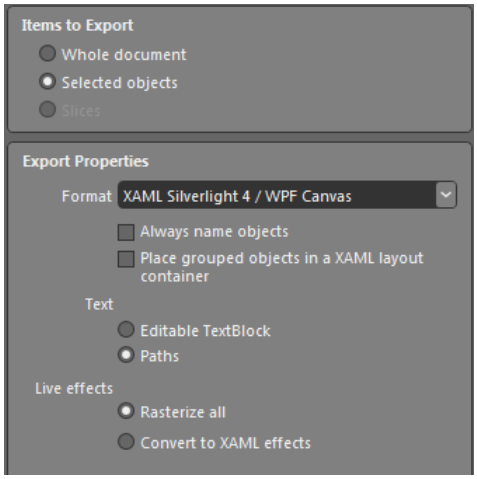
```
//Stroke Thickness
StrokeThickness="{Binding StrokeThickness, RelativeSource={RelativeSource ▶
AncestorType={x:Type signatures:ISymbolViewboxSignature}}}"
```

To create and import an XAML component with dynamic properties:

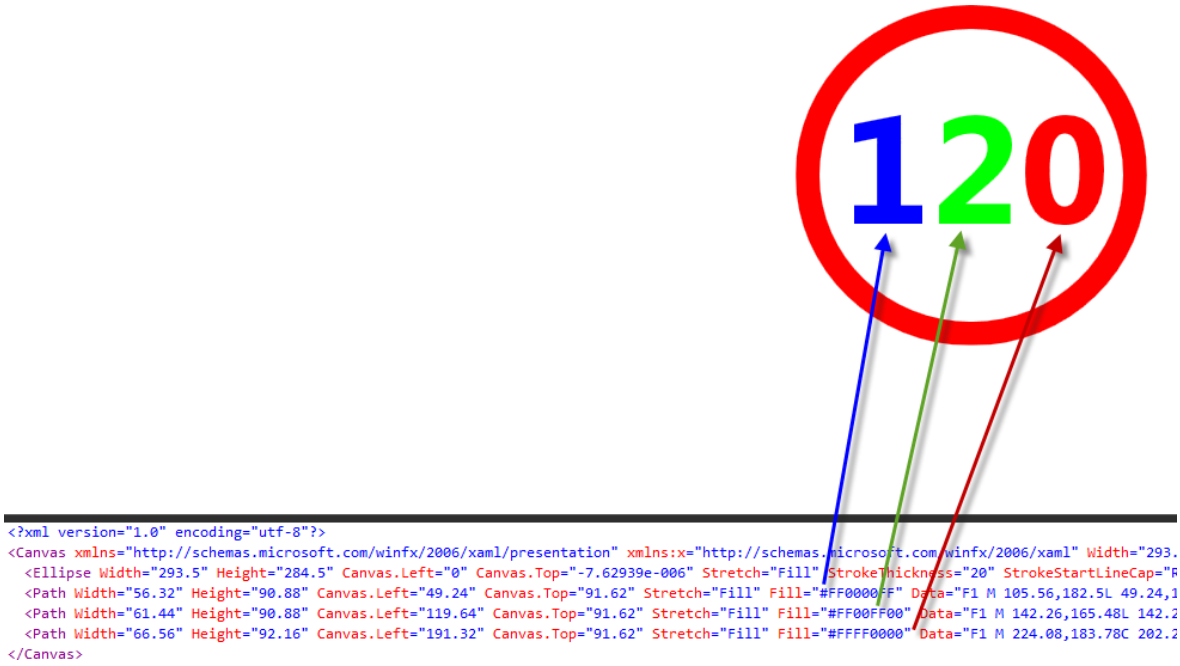
1. Download and install **Inkscape** from the Inkscape website.
2. Open **Expression Design** and create your graphics.



3. Select all objects and choose **File** > **Export**. Use the settings shown in the screendump below and click **Export All**.



4. Install **Kaxaml Tool**, a free tool for editing and previewing XAML files.
5. Open the XAML file in **Kaxaml Tool**.
6. Identify the elements to control with dynamic properties.
7. Temporarily change fill colors for easier identification.



8. Open iX Developer.
9. Right-click on a node in the **Component Library** in the Component Library Window
10. Select **Add Components** and choose your modified XAML file.

Your component will now reflect dynamic changes based on the values of **Fill**, **Outline**, and **Outline Thickness**.

> **NOTE**
> Always include the `signatures` namespace in your XAML file for dynamic properties to function correctly in iX Developer.

Example of XAML modification

Before (static properties):

```
<Ellipse Stroke="#FFFF0000" Fill="#FFFFFF00" StrokeThickness="20" ... />
```

After (with dynamic properties):

```
<Ellipse
    xmlns:signatures="http://www.beijerelectronics.com/hmi/ix/signatures"
    Stroke="{Binding Stroke, RelativeSource={RelativeSource ►
AncestorType={x:Type signatures:ISymbolViewboxSignature}}}"
    Fill="{Binding Fill, RelativeSource={RelativeSource ►
AncestorType={x:Type signatures:ISymbolViewboxSignature}}}"
    StrokeThickness="{Binding StrokeThickness, ►
RelativeSource={RelativeSource AncestorType={x:Type ►
signatures:ISymbolViewboxSignature}}}" ... />
```

11. Actions

In iX Developer, actions are predefined commands that can be triggered by user interactions or events within the application. They are essential for enhancing the functionality of screens and enabling dynamic user experiences.

11.1. Available Actions

The following actions are available from the **Select Action** dropdowns:

Screen

Action	Description
Close Screen	Closes the current screen. If triggered from a tag in a Popup Screen, both the popup and its parent screen are closed.
Print Screen	Saves the current screen as a `.png` file. **Options** • **USB**: Saves to a USB flash drive. • **SD Card**: Saves to an SD Card. • **Select at Runtime**: Allows the user to choose the save location in the iX App. • **AppData**: Saves to `ProgramData\OS3 Packages\<<Project-GUID>>\Exports\Screenshots`. **NOTE** If **Enable FTP Friendly Names** is enabled in the Advanced section of the **Project Properties**, spaces are replaced with underscores, and periods (except for before the extension) are replaced with dashes.
Show Next Screen	Displays the next screen.
Show Previous Screen	Returns to the previous screen.
Show Screen	Opens a specific screen, including customizable Popup Screen positions.
Show Start Screen	Displays the startup screen.

TIP
For more information on screens, see the Screens section.

Address Book

Action	Description
Open Address Book	Opens the address book for editing, typically used for Alarm Distributor purposes.

Alarm Distributor Server

Action	Description
Open Routes Configuration	Manages the routes for the distributed Alarm Server.

TIP
For more information on the alarm server, see the Alarm Server Function section.

Alarm Viewer

Action	Description
Acknowledge All Alarms*	Acknowledges all alarms across all Alarm Viewers.
Acknowledge Selected Alarms*	Acknowledges a selected alarm.
Acknowledge Visible Alarm*	Acknowledges only the visible alarms.
Alarm Info*	Executes the action configured for the requested alarm event info in the Alarm Server Function.
Clear All Alarms*	Removes all alarms in all Alarm Viewers.
Clear Visible Alarms*	Removes only visible alarms.
Enable/Disable selected Alarm*	Toggles the alarm state.
Filter Alarms*	Opens the filter configuration dialog.
Manage Alarms*	Adds the **Manage Alarms** button that opens the Manage Alarms Popup Window.
Pause Viewer*	Starts or pauses the Alarm Viewer.

* All **Alarm Viewer** actions must be associated with an Alarm Viewer in the same screen as the object with the action configured. This also applies to actions that affect all alarms.

TIP
For more information on the alarm viewer, see the Alarm Viewer section.

Chart

Action	Description
Pan Left/Right/Up/Down	Moves the chart view.

TIP
For more information on the chart object, see the Chart Object chapter.

Action	Description
Reset View	Resets zoom and pan settings.
Zoom In/Out	Adjusts the zoom level.

TIP
For more information on the chart object, see the Chart Object chapter.

Data Logger

Action	Description
Clear Datalogger	Clears the selected Data Logger.
Log Once	Logs a single entry in the Data Logger.
Start/Stop Logging	Toggles logging for the Data Logger.

TIP
For more information on data loggers, see the Data Logger Function section.

Database

Action	Description
Back Up Databases	Creates a backup of one or more databases. For more information, see Back Up Databases in the iX App.
Restore databases	Restores one or more databases. For more information, see Restore Databases in the iX App.
Database Cleanup	Optimizes performance by removing empty space and defragmenting the database. For more information, see Run Database Cleanup From the iX App
Database Export	Exports the database as a `.csv` file. For more information, see Export Database From the iX App.

TIP
For more information on databases, see the Databases section.

Output Devices

Action	Description
Email Configuration	Shows the e-mail configuration page. Used by the Alarm Distributor.

Recipe

Action	Description
Delete Recipe	Removes a Recipe.
End Offline Recipe Editing	Sets recipes to **Online mode**. For more information, see Edit Recipes Offline.
Export Recipe	Exports the selected recipe as a **.CSV** file. For more information, see Export Recipe Database From the iX App.
Import Recipe	Imports a recipe from a **.CSV** file.
Save/Load Recipe	Saves/Loads a Recipe.
Start Offline Recipe Editing	Sets recipes to **Offline mode**. For more information, see Edit Recipes Offline.

TIP
For more information on recipes, see the Recipe Function chapter.

Reporting

Action	Description
Generate Report	Creates a report file that can be printed or saved as an *Excel* file. **Options** • **USB**: Saves to a USB flash drive. • **SD Card**: Saves to an SD Card. • **Select at Runtime**: Allows the user to choose the save location in the iX App. • **AppData**: Saves to `ProgramData\OS3 Packages\<<Project-GUID>>\Exports\Reports`.  **TIP** If generating multiple reports, queue actions to reduce memory usage.  **NOTE** If **Enable FTP Friendly Names** is enabled in the Advanced section of the **Project Properties**, spaces are replaced with underscores, and periods (except for before the extension) are replaced with dashes.

TIP
For more information on recipes, see the Reports Function chapter.

Security

Action	Description
Export/Import User Accounts	Manages user accounts via file export/import.
Login	Opens the login dialog, see Login and Logout of the iX App.
Logout	Logs out the current user, see Login and Logout of the iX App.
Show Domain Login Configuration Wizard	Displays a wizard to configure domain login. For more information, see Domain Login.
Show Users Dialog	Displays user account management options, see Configure User Accounts.

TIP
For more information on security, see the Security section.

Settings

Action	Description
Change Active Controllers	Defines active Controllers. • The **IsActive** property can be modified by script for non-OPC UA controllers (e.g., `Globals.Controller1.IsActive = true;`). • For **OPC UA controllers**, a project restart is required for changes. • If the driver settings dialog, **Change Active Controllers** > **Settings**, is opened in the iX App, the controller enters **configuration mode** and becomes inactive until the settings are confirmed or the dialog is closed.
Disable/Enable Debug Logger	Turns debug logging on or off.
Set Language	Changes the interface language on the HMI panel. For more information on languages, see the Language and Translation section.
Set Time Zone, Region and Daylight Saving	Adjusts time zone, region, and daylight saving settings on the HMI panel
Show System Settings	Opens the OS3 menu.

Tag

Action	Description
Clear Non-Volatile Values	Clears stored values of non-volatile tags.

** This action can be used with Aliases.

TIP
For more information on tags, see the Tags section.

Action	Description
Decrement/Increment Analog**	Adjusts the value of an analog tag by a set amount.
Reset Tag**	Resets a tag value to 0.
Set Analog**	Assigns a specific value to an analog tag.
Set String**	Assigns a string value to a string tag.
Set Tag**	Sets a tag value to 1.
Toggle Tag**	Toggles a tag value between 0 and 1.

** This action can be used with Aliases.

TIP
For more information on tags, see the Tags section.

Trend Viewer

Action	Description
Show Trend Viewer Legend	Displays the legend in the selected Trend Viewer Object.
Trend Viewer History	Controls the display of historical or real-time trend data. **Options** • **On**: Displays history from the trigger moment. • **Off**: Switches back to real-time data. • **Toggle**: Switches between history and real-time modes. • **Show Dialog**: Allows the user to select a specific time range.

TIP
For more information on the trend viewer, see the Trend Viewer Object chapter.

Other

Action	Description
Run Script	Executes a script method.
Close Application	Closes the iX App.
Export Debug Log to External Storage	Saves the debug log to a file. **Options** • **USB**: Saves to a USB flash drive. • **SD Card**: Saves to an SD Card. • **Select at Runtime**: Allows the user to choose the save location in the iX App. • **AppData**: Saves to `ProgramData\OS3 Packages\<<Project-GUID>>\Exports`.

11.2. Action Triggers

An **Action trigger** is an event or condition that initiates a specific action or sequence of actions within the iX App. Action triggers enable predefined operations based on conditions such as alarms, tag value changes, or user interactions.

To view available action triggers for an object, select the object and navigate to the Actions Ribbon Tab.

BEST PRACTICES

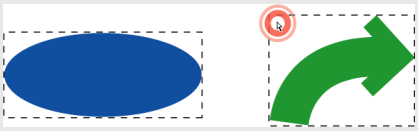
- **Avoid multiple triggers**: Using multiple trigger methods simultaneously can lead to unintended behavior.
- **Use scripts or program actions**: Instead of combining different trigger methods (e.g., Click Action Trigger with Mouse Button Action Triggers), use scripts or program actions from the Actions Ribbon Tab to ensure stable functionality.

11.2.1. Mouse Button Action Triggers

Action trigger	Description
Mouse Down	Executes an action when the left mouse button is pressed down on the object in the iX App.
Mouse Up	Executes an action when the left mouse button is released from the object in the iX App.
Mouse Enter	Executes an action when the cursor enters the boundaries of the object in the iX App.
Mouse Leave	Executes an action when the cursor leaves the boundaries of the object in the iX App.

NOTE

For most target types, the object boundaries are not adjusted to the visible outline of the shape. Instead, the active area extends to a rectangular bounding box. This means that mouse actions are triggered when the mouse pointer enters this rectangular bounding box.



11.2.2. Function Key Action Triggers

Action trigger	Description
Key Down	Executes an action when a function key is pressed in the iX App.
Key Up	Executes an action when a function key is released in the iX App.

11.2.3. Value Changed Action Triggers

Action trigger	Description
Input Value Changed	Executes an action when a new value is entered by the user.
Value Changed	Executes an action when the value of the object changes.

11.2.4. Tags Action Triggers

Action trigger	Description
Value Changed	The tag remains active at all times, triggering the action whenever the tag's value changes.
Value Changed	The tag is only active when required. If the tag is deactivated (i.e., not in use), the event remains idle until the tag is reactivated.

11.2.5. Focus Action Triggers

Action trigger	Description
Got Focus	Executes an action when the object is selected.
Lost Focus	Executes an action when the object is no longer selected.

11.2.6. Navigation Action Triggers

Action trigger	Description
Navigated	Executes an action when navigation to the address entered in the address field of the Web Browser Object is complete.
Navigating	Executes an action when an address has been entered in the Web Browser Object's address field but has not yet been reached.

11.2.7. Data Logger Action Triggers

Data Logger Action Triggers are used to execute actions based on data logging events.

NOTE
These triggers are not available in the Actions Ribbon Tab. Instead, they can be configured in the Data Logger Properties.

11.2.8. Alarm Server Action Triggers

Alarm server actions can be configured for the entire alarm server, alarm groups, or individual alarm items. These actions are activated when there are changes in alarm status.

Alarm Server Action Triggers execute actions based on changes in alarm status. These actions can be configured for:

- The entire alarm server
- Specific alarm groups
- Individual alarm items

NOTE
These triggers are not available in the Actions Ribbon Tab. Instead, they can be configured in theAlarm Server Properties Window.

11.3. Add an Action to an Object

Instructions

1. Select the object and go to the Actions Ribbon Tab. All available Action Triggers will be displayed.
2. Click **Select Action** in the desired action trigger group and choose an action from the Available Actions in the dropdown menu.

11.4. Login and Logout of the iX App

11.4.1. Login

Use the **Login** action to open the **Login** dialog in the iX App. For setup instructions, see **Security** > **Login** in Available Actions and Add a New Action to the Action Menu Object.

The dialog lists all users configured in the project, except those in groups marked as **Users Invisible in Runtime**, see Define and Assign Security Groups. Users can also change their password from this dialog.

You can configure the app to show the **Login** dialog automatically when a user tries to access a restricted object. This is configured in the Security Properties Dialog under: **General** > **Default settings on Access Denied** > **Select action on access denied** > **Show Login Dialog**.

11.4.2. Logout

Use the **Logout** action to log the user out of the iX App. For setup instructions, see **Security** > **Logout** in Available Actions and Add a New Action to the Action Menu Object.

To log out users after a period of inactivity, enable automatic logout in the Security Properties Dialog under: **General** > **Default settings on Access Denied** > **Automatic logout**.

After logout, the **Show Previous Screen** action (for function or touch keys) is disabled to block access to password-protected screens.

11.5. Set Up Actions Triggered by Tag Value Changes

To configure Actions to be triggered when a tag value changes:

1. In the Tags Configuration Page, go to the **Tags** tab.
2. Select **Columns Visible** > **Others**.
3. In the tag list, select the tag you want to use. In the **Actions** column, click **...** to open the **Actions Properties** dialog.
4. Select **Trigger** for the action.
5. Click **Add**, set the condition, and then click **OK**.
   - Use integers for numeric values.
   - For BOOL and STRING types, only == and != are supported.

6. Configure the action to run when the condition is true. See Add an Action to an Object for details.

> **NOTE**
> Do not set an action that changes the value of its own tag. This may cause unexpected behavior.

11.6. Configure Multiple Actions

To configure multiple actions for an object:

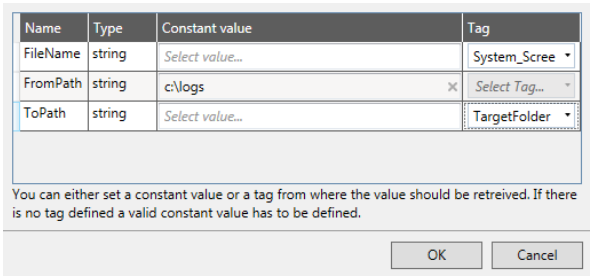
1. Select the desired object and go to the Actions Ribbon Tab.
2. Click the small arrow in the lower right corner of the desired Action Trigger group to open the **Actions Properties** dialog.
3. In the **Action Properties** dialog, configure and sort additional actions. The available actions depend on the selected object.

Once multiple actions are configured, this will be indicated for the action group on the Actions Ribbon Tab.

11.7. Apply Script Action

Script actions execute script methods from script modules based on triggered actions. To apply a script to a selected object:

1. Select the object and go to the Actions Ribbon Tab. All available Action Triggers will be displayed.
2. In the appropriate action trigger group, click **Select Action**
3. Select **Other** > **Run Script**.
4. Choose a **script module** from the available options.
5. Select a **script method** from the list of methods available for that script module.
6. Enter tags or fixed values for each script parameter in the dialog, and click **OK**.



The script action will execute when the object is triggered.

> **NOTE**
> Script actions take precedence over other script code for an object.

TIP
For detailed information about scripting in iX Developer, refer to the Scripts section.

12. Functions

Functions are essential components that enhance application development efficiency and organization by allowing the decomposition of complex processes into simpler, manageable parts. They can accept input values, perform specified actions, and return results, which streamlines coding and optimizes application performance.

Functions are particularly useful for:

- **Adding custom features.**
- **Automating tasks.**
- **Enhancing user interaction.**

You can find functions in the Functions Section of the Project Explorer Window.

This chapter will introduce you to functions, covering their basic structure, how to use them, and best practices for effective application development.

12.1. The Configuration Pages

Most functions are configured from their **configuration pages**. To access a configuration page, click the desired function in the Functions Section. Some features have multiple configuration pages.

TIP

For a list of keyboard shortcuts, see Keyboard Shortcuts.

12.1.1. Filter Items on the Configuration Pages

Use the **Filter Editor** to filter items on the The Configuration Pages.

Apply a filter with filter editor

1. Click **Show Selection** on the configuration page.
2. Use the operators and operands to apply a filter and click **OK**.
   - **Red**: Logical operator
   - **Orange**: Column operand
   - **Blue**: Value operator
   - **Green**: Value operand
3. An active filter appears at the bottom of the configuration page. Use the **x** to remove the filter and the **checkbox** to deactivate/activate the filter.

Additional information

- Filter Editor settings are context-sensitive.
- Name strings are not case-sensitive.
- Each configuration page saves its own filter settings during the editing session.
- Filters are structured as logical expression trees:
  - **Leaf nodes** = column value operations.

- **Main nodes** = logical operators.

12.2. Alarm Distributor Function

The **Alarm Distributor** function is used to send alarm notifications via email.

To add this function to the Functions Section, click the function in **Insert** > **Functions**.

TIP
For more details on alarms and alarm distribution, see the Alarms section.

NOTE
For proper functionality, ensure that the **Alarm Distributor** function is created before setting up any related actions.

12.2.1. Alarm Distributor Configuration Page

The **Alarm Distributor** configuration page is used to set up and manage alarm distribution settings, including receivers, filters, and schedules.

To open this page, click on the function in the Functions Section.

TIP
For more details on alarms and alarm distribution, see the Alarms section.

12.2.1.1. Route Section

A route defines filtering rules for distributing alarms. Routes allow notifications to be sent to different receivers. Click **Add** to add a route and **Delete** to delete a route.

12.2.1.2. Configure Distribution Devices

Click **Configure Distribution Devices** to open the Configure Distribution Devices Window.

12.2.1.3. Receivers Tab

This tab allows you to specify receivers for each route.

Parameter	Description
Address Book	Opens the **Address Book**, where you can add or edit email addresses of receivers.
Receiver Selection	Opens the **Receiver Selection** window, where you select receivers for an alarm route.
Parallel Receivers	Lists the selected receivers from the **Receiver Selection** window.

12.2.1.4. Alarm Filter Tab

This tab is used to configure filters for distributing alarms.

Parameter	Description
Alarm Name	Distributes alarm with this name.
Alarm Group	Distributes alarms in this group.
Alarm Text	Distributes alarms containing this specific text.
Alarm Status(es) to distribute	Distributes alarms with this status. At least one alarm status must be selected.

NOTE

- Filtering uses **"AND" logic**, meaning an alarm is distributed only if **all** specified conditions are met.
- An **empty string** acts as a **wildcard**.

12.2.1.5. Scheduler Tab

This tab is used to schedule alarm distributions based on time.

Parameter	Description
Period	Selects **Daily** or **Weekly** alarm distribution.
Day *(only for **Weekly** period)*	Selects a weekday for alarm distribution.
Start/Stop Time	Selects start and stop times for alarm distribution. Only alarms that occur **between** the start and stop times will be distributed.
Name	Auto-generated based on start and stop times. Cannot be edited.

12.2.1.6. Configure Distribution Devices Window

The **Configure Distribution Devices** dialog is opened from the **Configure Distribution Devices** button on the Alarm Distributor Configuration Page,

Settings

Parameter	Description
From Name	The sender's name that will appear on the email.
From Email Address	The sender's email address.
SMTP Server Name/IP	The SMTP server's name or IP address used to send the email.
Port	The port number of the email server.
Encryption Mode	Choose between **None**, **ImplicitSSL**, or **ExplicitSSL** for email encryption.
Advanced Encryption	Enable to use advanced encryption methods like DSS, DHE, and DH. DIsable to use only RSA encryption.
Code Page	Defines the character set for a specific language. Default code page in iX Developer is 65001 (UTF-8).

Parameter	Description
Authentication Mode	Select **None** or **Authenticated Login** for authentication.
Retry Sending	The number of retry attempts if the email cannot be delivered.
Time Interval	The interval (in seconds) between retries.
Subject	The subject of the email. Alarm Variables can be used.
Body	The body content of the email. Alarm Variables can be used.

12.2.1.7. Alarm Variables

Variables collected from alarm server definitions can be included in distributed alarm information. Variables are enclosed in curly brackets, for example:

```
"Alarm text: {3}"
```

Available alarm variables

Number	Variable	Description
0	State	The current state of the alarm.
1	StateTime	The time the alarm entered the current state.
2	AlarmId	A unique alarm ID.
3	AlarmText	Alarm item text.
4	AlarmItemDisplay- Name	Alarm item name.
5	Count	The number of times the alarm has occurred.
6	ActiveTime	The time the alarm became active.
7	InactiveTime	The time the alarm became inactive.
8	NormalTime	The time the alarm became normal.
9	AcknowledgeTime	The time the alarm was acknowledged.
10	AlarmGroupName	Alarm group name.
11	AlarmGroupText	Alarm group text.

12.3. Alarm Server Function

The **Alarm Server** function is used to configure alarm servers.

TIP
For detailed information on how to work with alarms and alarm servers, see the Alarms chapter.

12.3.1. Alarm Server Configuration Page

To open this page, click on the function in the Functions Section.

TIP
For detailed information on how to work with alarms and alarm servers, see the Alarms chapter.

Alarm Items tab

Parameter	Description
Name	Defines the symbolic name for the alarm item, displayed in the Alarm Viewer Object.
Text	Optional text shown in the Alarm Viewer Object. To make it dynamic, click **...** and configure the settings. . Can also be linked to the Text Library Function.
Tag	Sets the tag (digital or analog) that triggers the alarm when it reaches the specified status.
Expression	See Expressions for more details.
Condition	Defines the alarm trigger condition: • **Equal to/Not equal to** - Triggers when tag value is equal/not equal to **Trigger Value**. • **Greater than/Less than** - Triggers when tag value is greater/less than **Trigger Value**. • **Equal to greater than/Equal to less than** - Triggers when tag value is equal to or greater/less than **Trigger Value**. • **Rising/Falling Edge** - Triggers when the tag changes to one/zero.
Trigger Value	Specifies the numeric value that triggers the alarm based on the condition: `<tag value> <condition> <trigger value> = true`.
History	Determines when the alarm is removed from the alarm list. If checked, the alarm remains in the list until full. If unchecked, it is removed when acknowledged and inactive. If **Acknowledge Required** is unchecked, it is removed once inactive.
Acknowledge Required	Specifies whether the alarm must be acknowledged. If checked, the alarm requires acknowledgment before returning to normal. If unchecked, it returns to normal automatically once inactive.
Remote Acknowledge	Adds a digital tag that acknowledges all alarms when set to `1`.
Remote Ack Expression	See Expressions for more details.
Enable Distribution *(available only if alarm distribution is enabled in the Alarm Server Properties Window)*	Enables alarm distribution via email. If enabled for an alarm group, all alarms in that group are distributed.
Repeat Count	Limits multiple entries for a repeatedly triggered alarm. The alarm will appear only once in the alarm list, with the number of triggers included in the alarm text. A **Count** column can also be displayed in the Alarm Viewer Object. Active time can be shown for the first or last trigger.
Action	Configures one or more actions when the alarm is triggered.

Alarm Groups tab

Parameter	Description
Name	Defines an optional name for the alarm group.
Text	Sets an optional text for the alarm group.
Colors	Defines foreground and background colors for each alarm status.
Remote Acknowledge	Sets a digital tag that acknowledges all alarms in the group when set to 1.
Remote Ack Expression	See Expressions for more details.
Enable Distribution *(available only if alarm distribution is enabled in the Alarm Server Properties Window)*	Enables distribution of the alarms in the group.
Action	Configures one or more actions when a particular alarm event occurs within the group.

12.3.1.1. Alarm Server Properties Window

Open the **Alarm Server Properties** window by clicking **Settings** in the Alarm Server Configuration Page.

Parameter			Description
Actions	**Trigger**	**Select Trigger**	See Alarm Server Action Triggers for available triggers.
Alarm Distribution	**Enable Sending To Alarm Distributor Server**		Enable to forward alarms to the Alarm Distributor Function.
General	**Settings**	**Tag for remote acknowledge**	Remotely acknowledges all current alarms when the tag value is 1.
		Tag to clear all alarms	Clears all alarms from the server when the tag value is 1.
		Tag to enable/disable alarm server	Enables the alarm server when the tag value is 1.
	Show Alarm Indicator	**Active**	Shows indicator when an alarm is active.
		Inactive	Shows indicator when an alarm is inactive.
		Acknowledged	Shows indicator when an alarm is acknowledged.
	Log Settings	**Max Number of Logged Rows**	Limits the number of saved alarms. Oldest entries are deleted when limit is reached. Must fit available memory in the HMI panel.
	Repeat Counter	**Show the time of first occurrence**	Consolidates recurring alarms into a single entry. Displays first and last occurrence times.
		Show the time of last occurrence	

Parameter		Description
Server Mode	**Server Mode**	**Local**: Alarm server is active. Alarms are stored locally on the panel. **Disabled**: Alarm server is disabled. No alarms are stored or triggered.

12.3.1.2. Alarm Server Action Triggers

Actions can be triggered by various alarm server events. The following action triggers are available:

Action trigger	Description
Alarm acknowledge	Triggered once for every acknowledged alarm.
Alarm active	Triggered once for every alarm that becomes active.
Alarm deleted	Triggered once when the **Clear** button in the Alarm Viewer Object is pressed, if at least one alarm is deleted.
Alarm event info requested	Triggered when the **Info** button in the Alarm Viewer Object is pressed. Applied first to individual alarms, then groups, then server.
Alarm inactive	Triggered once for each alarm that no longer meets its alarm condition.
Alarm normal	Triggered once for each acknowledged alarm whose condition is no longer met.
Active alarms deleted	Triggered when active alarms are deleted due to exceeding the maximum logged rows.
Alarms changed	Triggered when multiple alarms change status at once. A collection of alarms is returned. Used primarily for scripting to avoid performance loss.
Alarms deleted	Triggered when multiple alarms are deleted at once via the **Clear** button in the Alarm Viewer Object. Used primarily for scripting to avoid performance loss.
Any acknowledged	Triggered when the alarm server toggles between having acknowledged alarms or not.
Any active	Triggered when the alarm server toggles between having active alarms or not.
Any inactive	Triggered when the alarm server toggles between having inactive alarms or not.

12.3.2. Export the Alarm Server Database From the iX App

Refer to the instructions in Export Database From the iX App.

12.4. Audit Trail Function

The **Audit Trail** function is a database that tracks and logs all operator actions that happens on the HMI panel. For all actions logged, refer to Actions Logged by the Audit Trail.

TIP

The Audit Trail Viewer Object displays the information from the audit trail database.

12.4.1. Audit Trail Configuration Page

The **Audit Trail** configuration page allows you to set up logging options for various actions. You can select whether to log all actions, specific actions, or only changes in tag values.

To open this page, click on the function in the Functions Section.

Parameter	Description
Log All/None	Logs all or no actions from the list.
Logging strategy	Sets your preferred logging strategy. Available options are FDA and Cyclic Buffering.
Max size of database	Specifies the maximum allowed size for the Audit Trail database.

12.4.2. How to Use the Audit Trail Function

12.4.2.1. Actions Logged by the Audit Trail

- Acknowledge All Alarms
- Acknowledge Selected Alarm
- Acknowledge Visible Alarms
- Backup Database
- Change Active Controllers
- Clear All Alarms
- Clear Data Logger
- Clear Non-volatile Values
- Clear Visible Alarms
- Copy Debug Log to USB Memory
- Database Cleanup
- Database Export
- Delete Recipe
- Decrement Analog
- Disable Debug Logger
- Email Configuration
- Enable Debug Logger
- Enable/Disable Alarms
- End Offline Recipe Editing
- Export Recipe
- Export User Accounts
- Generate Report
- Import Recipe
- Import User Accounts
- Increment Analog
- Login
- Log Once

- Logout
- Open Address Book
- Open Routes Configurations
- Pan Down
- Pan Left
- Pan Right
- Pan Up
- Print Screen
- Restore Database
- Run
- Run Script
- Save Recipe
- Set Analog
- Set Date and Time
- Set Language
- Set String
- Set Tag
- Set Time Zone, Region, and Daylight Saving
- Show Backlight Settings
- Show IP Settings
- Show Next Screen
- Show Previous Screen
- Show Screen
- Show Unit Conversion Dialog
- Show User Dialog
- Start Logging
- Start Offline Recipe Editing
- Stop Logging
- Storage Device Detection
- Trend Viewer History
- Toggle Tag
- Zoom In
- Zoom Out

12.4.2.2. Configure Audit Trail Descriptions

You can add, view, or change a description for each object recorded in the Audit Trail database. The description can be up to 255 characters long.

Instructions

1. Navigate to the Tags Function Configuration Page,
2. Enable **Columns Visible** > **Others**.

3. In the **Description** column, you can view, add, or modify the audit trail description.

12.4.2.3. Export the Audit Trail Database From the iX App

Refer to the instructions in Export Database From the iX App.

> **NOTE**
> Exporting the database will not delete the audit trail log from the project.

12.4.2.4. Enable Audit Trail Logging for Tag Changes

To log changes in tag values made by user actions:

1. Navigate to the Tags Configuration Page,
2. Enable **Columns Visible** > **Others**,
3. Select **Log to Audit Trail** from the list.

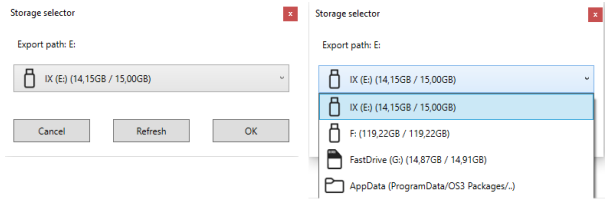
> **NOTE**
> This option is only available when the Audit Trail Function is enabled in the project.

12.4.2.5. Logging Strategies

Two logging strategies are available:

FDA

- **Compliance**: Ensures adherence to FDA (Food and Drug Administration) requirements.
- **Overwrite behavior**: Log entries are never overwritten, ensuring complete data integrity.
- **Database limit**: When the database reaches about 80% capacity, a notification will prompt you to export the data.
- **Database size**: You can increase the size of the Audit Trail database if necessary.
- **Backup**: If the database limit is exceeded and a USB drive is connected to the HMI panel, the system will automatically back up the data to the USB drive with the lowest alphabetical designation.
- **Storage options**: If no USB is inserted, a pop-up menu will prompt you to select a storage location: **SD card**, **USB**, or **AppData** (the `ProgramData` folder). If new storage media is inserted, tap **Refresh** to update available options.



Cyclic Buffering

- **Overwrite behavior**: The oldest log entries are overwritten once the database reaches its capacity.

- **No warning**: No warning will appear when the database limit is reached.
- **Export available**: Regardless of the strategy used, the database can still be exported. Log entries remain in the Audit Trail database and are accessible via the Audit Trail Viewer Object.

12.5. Data Logger Function

The **Data Logger** function allows you to log and save data in the iX database. You can connect any number of tags to a data logger, enabling you to log data values of various types. Multiple data loggers can be used within a project, each with different sampling options.

To add this function to the Functions Section, click the function in **Insert** > **Functions**.

TIP

- For detailed information about databases, refer to the Databases section.
- For instructions on how to optimize data logger usage, see Data Logger Performance.

12.5.1. Data Logger Configuration Page

To open this page, click on the function in the Functions Section.

Parameter	Description
Add/Delete	Adds or deletes a **SchedulerJob**.
Settings	Opens the Data Logger Properties.
Show Selection	Opens the **Filter Editor**, see Filter Items on the Configuration Pages.
Name	Symbolic name for the logged tag.  **NOTE** **Time** or **Id** cannot be used as names because these are reserved.
Tag	The tag to log.
Expression	See Expressions for more details

12.5.1.1. Data Logger Properties

Parameter			Description
Action	**Trigger**	**Log item Value changed**	Triggers when the value of a specific log item changes.
		Log values updated	Triggers when any log value is updated.
Settings	**Log with interval**		Sets a time interval (in seconds) to collect frequently changing data. The minimum interval is 1 second.

Parameter		Description
	Log on tag event	Select this option to control data sampling based on events rather than a regular time interval. Choose the tag to use as a trigger. Data will be sampled when the tag value changes.
	Log changes only	When enabled, data will be logged only if the value differs from the last logged value. Enable this option to save memory.
Log Settings	**Max number of logged rows**	Defines the maximum number of tag values that can be saved in the database. When this limit is reached, the oldest item will be overwritten. Ensure the value is within the available memory capacity of the HMI panel.

12.6. Function Keys Function

The **Function Keys** function is used to assign Actions or Scripts to the function keys, **F1** - **F12**.

To add this function to the Functions Section, click the function in **Insert** > **Functions**.

12.6.1. Function Keys Configuration Page

To open this page, click on the function in the Functions Section.

Parameter	Description
Function Key Context	• **Global**: The function key behaves the same across all screens. It is accessible whenever the HMI panel is running, unless the current screen has its own function key setup. Screen-specific keys take precedence over global ones, while background screen keys have lower priority than screen-specific keys but higher than global keys. • **Screen**: The function key affects only the current screen.
Actions	Sets the actions to be performed when the function key is pressed or released. You can configure multiple actions for the same key. For more details, see Multiple Actions.
Security groups required	Specifies which Security Groups can access the function key.

12.6.2. Configure Function Keys

Function keys can be configured to run actions or scripts.

Function key limitations

- Function keys are disabled when HMI panel backlight is disabled.
- Only two function keys can be activated at once. If more than two are pressed simultaneously, only the first two will function. To execute more actions, consider using buttons or other on-screen objects connected to the desired actions.

12.6.2.1. Configure a Function Key to Run Actions

Instructions

1. Click on **Project Explorer** > **Functions** > **Function Keys** to access the Function Keys Configuration Page.
2. Choose either **Global** or **Screen** for the function key's context. Refer to Function Keys Configuration Page for details.
3. Click **Add** and select the function key you want to define.
4. In the **Actions** column, click **...** to configure the actions performed when the function key is pressed or released. You can configure multiple actions for the same key. For more information, see Multiple Actions.
5. In the **Security groups required** column, specify which Security Groups can access the function key.

12.6.2.2. Configure a Function Key to Run Scripts

Instructions

1. Click on **Project Explorer** > **Functions** > **Function Keys** to access the Function Keys Configuration Page.
2. Choose either **Global** or **Screen** for the function key's context. Refer to Function Keys Configuration Page for details.
3. Click **Add** and select the function key you want to define.
4. In the lower part of the Editor Area, select **Script** to switch to Script View Mode.
5. In the left menu, click the **+** next to the desired function key.
6. Double-click **KeyDown** and enter the script code you want to execute when the function key is pressed.

Example

The following C# code sets function key **F12** to acknowledge all active alarms:

```
public partial class FunctionKeys
{
    void F12_KeyDown(System.Object sender, System.EventArgs e)
    {
        Globals.AlarmServer.Acknowledge();
    }
}
```

TIP

For detailed information on scripting in iX Developer, refer to the Scripts section.

12.6.2.3. Configure a Function Key with Momentary Function

This configuration keeps the tag active only while the function key is pressed, creating a momentary effect.

Instructions

1. Click on **Project Explorer** > **Functions** > **Function Keys** to access the Function Keys Configuration Page.
2. Choose either **Global** or **Screen** for the function key's context. Refer to Function Keys Configuration Page for details.
3. Click **Add** and choose the function key you want to configure.
4. Set a **Set Tag** action for **Key Down** to activate the tag when the key is pressed.
5. Set a **Reset Tag** action for **Key Up** to deactivate the tag when the key is released.

12.6.2.4. Function Key Behavior During Screen Transitions

The behavior of function keys during screen transitions depends on the **Function Key Context** on the Function Keys Configuration Page and the screen settings.

NOTE
Screen function keys take priority over **Global** function keys.

Screen transition scenario	Result
Screen function key is pressed, and the screen changes.	• The **up** action for the local key is triggered. • No additional actions occur on the new screen while the key remains pressed.
Global function key is pressed, and the screen changes to one **without** a local setting for the same key.	• The **up** action for the global key is triggered when the key is released. • No action occurs during the screen transition.
Global function key is pressed, and the screen changes to one **with** a local setting for the same key.	• The **up** action for the global key is triggered. • No additional down or up actions are triggered on the new screen while the key remains pressed.

12.7. Multiple Languages Function

You can translate texts and system texts directly in the application or by exporting them to a text file for translation in external software. After translation, re-import the file into the application. The application language can be switched in the iX App such as by using a tag value. For more details, see the Language and Translation section.

12.7.1. Multiple Languages Configuration Page

To open this page, click on the function in the Functions Section.

12.7.1.1. Texts Tab

The **Texts** tab displays texts added to objects by the developer. You can also edit texts in the default language here.

If an object is linked to a text from the Text Library Function, the object's internal denomination (not displayed in the iX App) are shown in grey in the **Texts** list. This label indicates that editing will not affect the iX App.

12.7.1.2. System Texts Tab

In the **System Texts** tab, you can edit system texts. Pre-installed translations are available for:

- English
- German
- French
- Brazilian Portuguese
- Chinese (Traditional and Simplified)

12.7.1.3. Languages Tab

The **Languages** tab allows you to manage languages. To add new languages and configure multiple language options, see Set Up Multiple Languages.

12.7.1.4. Text ID Tab

When **Text IDs** are enabled, all text box controls in iX Developer, including those in the iX App, will be replaced with the **Text ID Browser** control. For more information, see Text IDs.

The **Text ID** tab contains one column for each additional language.

12.8. Recipe Function

Recipes make it possible for a set of tags to be saved in the HMI panel to be downloaded at a later time. The operator can download a saved recipe to the controller, which will start working with the new values. Recipe libraries consisting of recipes with different sets of parameters can be created, allowing reuse of large sets of parameters.

Recipes can be created during the design of the project or from the HMI panel.

Recipe data is stored in the HMI panel database. Recipe handling functions such as loading, storing, copying and deleting recipes are handled with Actions. Recipe data is not stored in a database during the design of the project. The recipe data is transferred to the target, and the database on the target is updated. This makes it possible to change a recipe without deleting any existing recipes when transferring the project to the target. The same applies for the Data Logger Function, changing the configuration of one data logger doesn't affect the other data loggers and recipes.

To add this function to the Functions Section, click the function in **Insert** > **Functions**.

12.8.1. Recipe Configuration Page

The **Recipe** function configuration pages manages tag sets and values via the **Tag Configuration** tab and the **Runtime Data** tab.

To open this page, click on the function in the Functions Section.

12.8.2. How to Work With Recipes

The following chapters describe how to work with recipes in iX Developer and the recipe actions that operators can perform in the iX App.

12.8.2.1. Load Recipe Action

The **Load Recipe** action transfers recipe values directly to the controller. This action can be used while the HMI panel is running. For more information about this action, see the Available Actions section.

12.8.2.2. Save Recipe Action

The **Save Recipe** action (see Available Actions) saves the current values of tags defined in the **Tag Configuration** tab on the Recipe Configuration Page. This action can be used while the HMI panel is running. For more information about this action, see the Available Actions section.

When the **Save Recipe** action is activated, the current tag values are automatically saved in the selected recipe file. If a recipe file with the specified name already exists, it will be overwritten during the save process.

The **Save Recipe** action can also be used to create new recipes in the iX App. If no existing recipe is selected for overwriting, the recipe data field will remain open. The operator will then be prompted to either save the recipe with a custom name or overwrite an existing one.

12.8.2.3. Add a Recipe to a Tag

Instructions

1. Open the Recipe Configuration Page.
2. Go to the **Tag Configuration** tab and click **Add** to create a **RecipeItem**.
3. In the **Tag** column, select a tag for the **RecipeItem**. The data type is already set based on the tag configuration.
4. Go to the **Runtime Data** tab and click **Add** to add a **RecipeTitle**.
5. Name the recipe title and enter values for each recipe item.
   Naming rules
   - A recipe can include up to 256 items.
   - The recipe name must be an alphanumeric string starting with a letter.
   - Use only letters (a-z, A-Z), numbers (0-9), and underscores (_) in the name.
   - Avoid using system device names (e.g., COM1, AUX) as recipe names. For more details, see Object Issues.

12.8.2.4. Edit Recipes Offline

You can set tags in recipes in **Offline mode** to edit existing recipes on the HMI panel without transferring values to the controller. In this mode, the tags in the selected recipe will not communicate with the controller.

The following actions control offline/online mode:

- **Start Offline Recipe Editing**: Puts the tags in the specified recipe in offline mode and displays a notification message.
- **End Offline Recipe Editing**: Returns the tags in the specified recipe to online mode, updating them from the controller.

For a complete list of all actions, see Available Actions.

NOTE
Internal tags can be used in recipes but are not affected by offline mode. While in offline mode, the Load Recipe Action and Save Recipe Action will not impact controller values.

Example procedure for editing recipes offline

1. Create a recipe according to the Add a Recipe to a Tag chapter.
2. Create a screen with four button objects.
3. Connect two of the buttons to the **Load Recipe** and **Save Recipe** actions.
4. Connect the other two buttons to the **Start Offline Recipe Editing** and **End Offline Recipe Editing** actions.
5. Run the project and test the functionality:
   a. Click **Start Offline Recipe Editing**. The tags are set in offline mode.
   b. Click **Load Recipe**. The tags are filled with data from the recipe.
   c. Change the data as needed.
   d. Click **Save Recipe**. The data in the tags is written to the recipe.
   e. Click **End Offline Recipe Editing**.

The tags are set in online mode and the values are updated with data from the controller.

12.8.2.5. Import and Export Recipes

This chapter describes how to import and export recipes in iX Developer and in the HMI panel.

Export Recipe in iX Developer

Instructions

1. Go to Recipe Configuration Page, navigate to the **Runtime Data** tab, click the small arrow next to the **Import** button and choose **Export**.
   Alternatively...
   right-click on the recipe in the Recipes Section and choose **Export Recipe**.
2. Select the recipe to export and specify file name.
3. Choose delimiter: **Comma** or **Semicolon**.
4. Choose file encoding: **Unicode (UTF-8)**, **Unicode** or **Unicode (Big-Endian)**.
5. Click **OK**.

This exports the recipe database as a **.csv** file.

WHEN EXPORTING TO EXCEL

When exporting a .csv file for editing in Excel, ensure the format aligns with Excel's regional settings based on the operating system's region settings.

- The default delimiter is set to match OS settings for compatibility with Excel's import/export functions.
- If the delimiter is changed and edited in Excel, the file may not reliably re-import into iX Developer.

Export Recipe Database From the iX App

You can export the database on the HMI panel as a **.csv** file and save it to an SD card, USB memory or in the project files folder. The database is exported via an action. For more information, see Actions.

> **NOTE**
> You cannot export databases during project simulation.

Instructions

1. In iX Developer, select the object the should trigger the export.
2. Go to the Actions Ribbon Tab.
3. On the the desired Action Trigger, click **Select Action** > **Recipe** > **Export Recipe**.
4. In the **Select a database** dropdown, select a recipe database.
5. Click **...** to open the **Export Recipe** settings.
6. Select the recipe to export and specify file name.
7. Choose delimiter: **Comma** or **Semicolon**.
8. Choose file encoding: **Unicode (UTF-8)**, **Unicode** or **Unicode (Big-Endian)**.
9. Click **OK**.
10. In the lowest dropdown, select export location:
    - **USB**: Saves to a USB flash drive.
    - **SD Card**: Saves to an SD Card.
    - **Select at Runtime**: Allows the user to choose the save location in the iX App.
    - **AppData**: Saves to `ProgramData\OS3 Packages\<<ProjectGUID>>\Exports`.

The object is now configures to export the recipe database when triggered.

> **NOTE**
> - If **Enable FTP Friendly Names** is enabled in the Advanced section of the **Project Properties**, spaces are replaced with underscores, and periods (except for before the extension) are replaced with dashes.
> - To display certain characters (such as Chinese or Arabic) correctly in Excel, select **Unicode (UTF-8)** in **File encoding**.

Import Recipe in iX Developer

Instructions

1. Go to the Recipe Configuration Page, navigate to the **Runtime Data** tab, and click **Import**.
   Alternatively...
   Right-click on the recipe in the Recipes Section of the Project Explorer Window, and click **Import Recipe**.
2. Configure the Import Recipe Settings and click **OK**.

Import Recipe Database to the HMI Panel

A recipe that previously has been exported as a **.csv** file, can be imported back into the HMI panels recipe database. The recipe is imported via an action. For more information, see Actions.

> **NOTE**
> Recipe import is not supported during project **simulation**.

Instructions

1. In iX Developer, select the object the should trigger the import.
2. Go to the Actions Ribbon Tab.
3. In the desired Action Trigger group, click **Select Action** > **Recipe** > **Import Recipe**.
4. In the **Select a database** dropdown, select a recipe database.
5. Click **...** to open the Import Recipe Settings. Configure the settings and click **OK**.

The object is now set up to import the database when triggered.

Import Recipe Settings

Recipe Runtime Data

Parameter		Description
Select a Recipe to import		Select which recipe to import.
Name of file to import from		Select the name of the file to import from.
Select the delimiter in the import data		Select the delimiter in the **.csv** file: **Comma** or **Semicolon**.
Import strategy when the item already exists in the database	**Ignore**	Select to ignore the new data.
	Replace	Select to replace the old data if the data already exists.

12.9. Reports Function

The report generator generates *Excel* reports based on an Excel made report template. The generated report can be printed, saved as an Excel file or as a simplified PDF file.

To add this function to the Functions Section, click the function in **Insert** > **Functions**.

12.9.1. Reports Configuration Page

To open this page, click on the function in the Functions Section.

Reports tab

Parameter	Description
Add/Delete	Adds or deletes a report.
Show Selection	Opens the **Filter Editor**, see Filter Items on the Configuration Pages.
Name	Symbolic name for the report.

Parameter	Description
File name	Name of the *Excel* report template.
Report format	Sets the format of the report: **Excel**, **PDF** or **Both Excel and PDF**.

Queries tab

Parameter	Description
Add/Delete	Adds or deletes an SQL query.
Show Selection	Opens the **Filter Editor**, see Filter Items on the Configuration Pages.
Name	Name for the SQL query.
Database	Sets the targeted database: **General** or **Audit**.
Query	The **SELECT SQL** query to be executed.

12.9.2. Add a Report

Instructions

1. Open the Reports Configuration Page.
2. Click **Add** and either type in or browse to the location of the *Excel* template file. Click **OK**.

NOTE

- If you make any changes to the report template after uploading it to the project, you must upload the template again.
- Ensure that each template file added to the project has a unique file name.

12.9.3. Add an SQL Query

Instructions

1. Open the Reports Configuration Page, and go to the **Queries** tab.
2. Click **Add**.
3. In the **Name** field, choose a unique name for the query.
4. In the **Database** field, select either **General** or **Audit**.
5. In the **Query** field, write your **SELECT SQL** query.

12.9.3.1. Include SQL Query Results in Reports

To include SQL query results in *Excel* reports:

- Define a **named range** in Excel with the format `QueryName`. The `QueryName` is the identifier for your SQL query.
- Use the syntax `<#QueryName.DatabaseColumnName>` in the report cells, where `DatabaseColumnName` is the column name from your SQL database.
- Named range cells can also be used in **Excel charts** and **functions**.
- To include the named range in functions, insert an **empty row** below the named range cells. This row must be included in the function.

12.9.4. Set up a Reports Template

To display tag data in a report file, use placeholders in the report template. Placeholders act as substitutes for tag data, which will be filled in when the report is generated in the iX App.

1. Check the Reports Template Limitations below to ensure compatibility.
2. In the *Excel* report template, enter the following placeholder text inside a cell:

   ```
   <#Tag(TagName)>
   ```

   Replace `TagName` with the actual tag name to be displayed.
3. Add the reports template to the iX Developer project, refer to Add a Report.

12.9.4.1. Reports Template Limitations

- **Microsoft Excel .xls format (2007) is not supported.**
- **Microsoft Excel .xlsx format supports up to 1,048,576 rows**—exceeding this may cause data loss.
- Report generation is **not instant**; delays may occur based on the number of tags processed.
- **Database-driven reports** are configured per project, not per Excel template. This means that if a user wants a database query on multiple projects, it has be configured on each project.
- **Tag placeholders** can only be used in Excel **cells**, not in elements like **WordArt**.
- **PDF conversion limitations**:
  - Diagrams are not supported.
  - Simplified Chinese and Korean are not supported.
  - Only portrait format is available.
  - If the content of an *Excel* cell is large enough to overflow the document, generating a PDF file will trim the data exceeding the page width.
- **Supported Excel features:**
  - **Text**: Helvetica, Times New Roman, Courier, Arial, Verdana, Symbol, Zapf Dingbats.
  - **Text size**
  - **Text formatting**: Size, Italic, Bold, Underline.
  - **Graphics**: Bitmap images.
  - **Cell formatting**: Borders, Background color.
  - **Page elements**: Header/Footer.
- **Query limitations:**
  - Queries must be written in the language supported by the selected database.
- **Report size limitation**
  - Reports generating more than 1 million cells of SQL query output may cause the iX App to run out of memory. For large reports, back up the database and use Transfer Client CLI to copy it to a PC with appropriate reporting tools.

12.10. Scheduler Function

The **Scheduler** function controls tags based on the real-time clock. This function is used to control process events, such as starting and stopping motors, at specific predefined calendar times.

To add this function to the Functions Section, click the function in **Insert** > **Functions**.

12.10.1. Scheduler Configuration Page

The **Scheduler** configuration page is used to set up scheduler jobs, allowing a set of tags to be activated based on specified start and stop times.

To open this page, click on the function in the Functions Section.

Parameter	Description
Add/Delete	Adds or deletes a **SchedulerJob**.
Show Selection	Opens the **Filter Editor**, see Filter Items on the Configuration Pages.
Name	The symbolic name for the scheduled event.
Period	Sets time interval for the event.
Start Date	Sets the date for the first scheduled event.
Start Time	Sets the time for the first scheduled event.
Duration	Sets the length of the event.
Tag	A **digital tag** set to `1` during the specified interval.
Stop Time Enable	Enables **Stop Date** and **Stop Time** settings.
Stop Date *(only used when **Stop Time Enable** is checked)*	The end date of the event. Resets the defined tag to `0`.
Stop Time *(only used when **Stop Time Enable** is checked)*	The end time of the event. Resets the defined tag to `0`.
Action	Configures an action to be performed when the scheduled event is **activated** or **deactivated**.

IMPORTANT

Do not use the same trigger tag for two different schedulers. When the first scheduler finishes, its trigger tag is reset to `0`, preventing the second scheduler from being triggered.

12.11. Security Function

The **Security** function allows you to create a security system for your project. Operators can be assigned different **authorization levels** to restrict access to specific **objects and functions**.

- **Security group information and passwords** are stored in the project database.
- This function is **optional** and does not need to be used if unrestricted access is acceptable.

TIP

For more details, refer to the Security section.

12.11.1. Security Configuration Page

To open this page, click on the function in the Functions Section.

TIP
For more details, refer to the Security section.

Users tab

Parameter	Description
Add/Delete	Adds or deletes a user.
Settings	Opens the Security Properties Dialog.
Show Selection	Opens the **Filter Editor**, see Filter Items on the Configuration Pages.
Import	Imports or exports an XML file.
Username	Any alphanumeric string, beginning with a letter (a-z, A-Z).
Password	Any alphanumeric string. Minimum length according to **Minimum password length** in the Security Properties Dialog. **IMPORTANT** Follow the Password Guidelines.
Description	Optional user description.
Groups	Sets the security groups for the user.

Groups tab

Parameter	Description
Add/Delete	Adds or deletes a security group.
Show Selection	Opens the **Filter Editor**, see Filter Items on the Configuration Pages.
Name	Any alphanumeric string, beginning with a letter (a-z, A-Z).
Users	The users defined in the **Users** tab to be included in the group.
Users Invisible in Runtime	When checked, users in the current group will not appear in the **Login** dialog in the iX App. However, users can still log in by manually entering their username and password. **TIP** To display an on-screen keyboard on an HMI panel to manually enter a username and password, click the **Login** header in the iX App login dialog, then click on the **User** field.

Domain Login tab

Parameter		Description
Domain Login Type	**Active Directory**	For details on **domain login** and setup instructions, refer to the Domain Login chapter.
	Microsoft Entra ID	

Parameter		Description
	Active Directory Federation Services	

12.11.1.1. Security Properties Dialog

The **Security Properties** dialog is opened from the **Settings** button on the Security Configuration Page.

Parameter			Description
General	**Default settings on Access Denied**	**Select action on access denied**	Sets an action for when a user's access rights are insufficient (e.g., when modifying an object or performing an action). Options include displaying an access denied message, opening a login dialog, or selecting **None** to perform no action.
		Select visibility	Specifies the visibility of objects for users with insufficient access rights: • **Disabled**: The object is visible but disabled for users with insufficient access rights. Only users with sufficient rights can interact with it. • **Hidden**: The object is completely invisible to users with insufficient access rights. Only users with sufficient rights can see it. • **Normal**: The object appears normal to users with insufficient access rights, but only users with sufficient rights can interact with it.
	Automatic logout		Enables automatic logout after a certain number of minutes of inactivity. After logout, the system will jump to the start screen unless access to the current screen is granted. If a communication error (**COM error**) is active, the logout will be delayed until communication is reestablished.
Password Rules	**Minimum password length**		Sets the required minimum number of characters for passwords.

12.12. Tags Function

Data values in a controller are referred to as **tags**. Tags may belong to the system or be internal. A tag has a symbolic name and can be of different data types. Objects connected to tags can change values in the controller, and tag values can be reflected by changing object appearance in various ways. Objects in a screen will remain static until connected to a tag.

TIP

For details on how to use tags, refer to the Tags section.

12.12.1. Tags Configuration Page

To open the **Tags** configuration page, click **Tags** in the Functions Section of the Project Explorer Window.

TIP
For details on how to use tags, refer to the Tags section.

12.12.1.1. Tags Tab

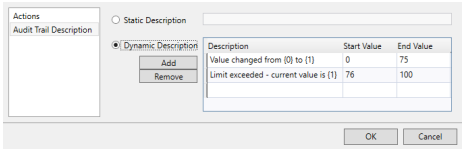
You can customize the view in **Tags** tab by selecting which columns to display in the **Columns Visible** section. Key settings remain visible at all times.

Columns visible

Columns Visible	Description
Scaling	Displays scaling columns for controller values. See Scaling.
Data Exchange	Displays the **Direction** and **When** columns for real-time data sharing triggered by digital tags or intervals set in **Tags** > **Triggers**. See Data Exchange.
Others	Displays additional tag settings.

Settings

Parameter		Description
Tag	**Name**	A symbolic name (alphanumeric, starts with a letter). Rename by clicking **...**.
	Data Type	Sets the format for engineering units when Scaling is applied. By default, it follows the controller's **Data Type** setting. For more information, see Tag Format and Data Type.
	Access Rights	Sets access permissions.
Controllers	**Data Type**	Selects the data type of the tag. For more information, see Tag Format and Data Type.
	Controller	Assigns the tag to a controller specified in the Controllers Tab. If unspecified, the tag is treated as internal.
Other	**Description**	Optional description (max. 200 characters). Auto-filled for system tags.
	Poll Group	Enables polling of tag groups at different intervals. See Poll Groups Tab and Poll Groups.
	Always Active	Keeps tag active for scripts or actions. Overridden by the **Active** setting in the Controllers Tab.
	Non Volatile *(internal tags only)*	Retains last value across power cycles (saved every 10 seconds).
	Initial Value *(internal tags only)*	Sets a default value at startup.
	Index Register	Assigns a register to read tag values. See Index Registers.

Parameter		Description
	Action	Configures tag-triggered actions. See Set Up Actions Triggered by Tag Value Changes.
	Log to Audit Trail *(if Audit Trail Function is activated)*	Tracks operator interactions.
	Audit Trail Description *(if Audit Trail Function is activated)*	Adds static/dynamic descriptions of actions for the Audit Trail Function, visible in the Audit Trail Viewer. When **Dynamic Description** is selected in the Audit Trail Viewer, use {0} for pre-change and {1} for post-change values. 
Scaling	**Offset**	See Scaling.
	Gain	See Scaling.
	Read Expression	See Scaling.
	Write Expression	See Scaling.
Data Exchange	**Direction**	See Data Exchange.
	When	See Data Exchange.

12.12.1.2. Controllers Tab

TIP

For details on **Controllers**, see the Controllers section.

Settings

Parameter	Description
Add	Adds a controller.
Delete	Deletes a controller.
Controller	Selects controller.
Settings	Opens controller-specific settings.
Show Selection	Displays filter options. See Filter Items on the Configuration Pages.
Name	Controller name.
ID	Optional short name shown in features such as Data Exchange.
Active	Activates the controller. Can also be controlled via script or action from the iX App.

12.12.1.3. Triggers Tab

Settings

Parameter	Description
Name	Symbolic name (alphanumeric, starts with a letter).
Tag	Sets the tag that, when its value changes to anything other than 0, triggers the data exchange.
Time	Defines interval using the format `-d.HH:MM:SS.ff` (days, hours, minutes, seconds, fractions of a second). A minus sign indicates a negative interval.

TIP
For more details on **Triggers**, see Triggers.

12.12.1.4. Poll Groups Tab

Settings

Parameter	Description
Name	Symbolic name (alphanumeric, starts with a letter).
Interval	Polling interval in milliseconds.

TIP
For more details on **Poll Groups**, see Poll Groups.

12.12.1.5. Index Registers Tab

Settings

Parameter	Description
Name	Index register name.
{Controller}	Each column corresponds to a controller listed in the Controllers Tab. Up to eight index registers can be defined per controller. Registers can be assigned to multiple objects.

TIP
For more details on **Index Registers**, see Index Registers.

12.12.1.6. Filter Tags

Use the **Filter Editor** on the Tags Configuration Page to display a specific selection of tags. For details on how to use the **Filter Editor**, see Filter Items on the Configuration Pages.

You can also filter tags by typing directly into the **Filter** field on the Tags Configuration Page. The tag list will update in real-time as you enter text.

12.13. Text Library Function

From the **Text Library** function, you can create text tables that link values to specific texts. This function is useful for displaying each sequence step in sequence control. Another use is to present error codes, where an analog signal triggers error codes linked to texts in the text library. Additionally, the function can assign specific values to analog signals based on the selected texts.

To add this function to the Functions Section, click the function in **Insert** > **Functions**.

12.13.1. Text Library Configuration Page

To open this page, click on the function in the Functions Section.

Parameter	Description
Add/Delete	Adds ore deletes text and group.
Show Selection	Opens the **Filter Editor**, see Filter Items on the Configuration Pages.
Import	Imports or exports text. The process is similar to Importing and Exporting Tags.
Text	Sets the text to be used.
Start Value	Defines the start and end values for text changes based on the tag's value.
End Value	

NOTE

- **Start Value** and **End Value** must be whole numbers. Decimal values may cause issues with text updates. Use Configure Texts Window if decimal values are required.
- A default text entered in Configure Texts Window will not be displayed outside the range set in the **Text Library**.

TIP

For instructions on how to connect the text on an object to a text library text, refer to Connecting Objects to Text Library Texts.

13. Alarms

Alarms notify users of events that require immediate action. An alarm is triggered when specific conditions are met, based on the logical evaluation of a tag value. Alarms can be organized into groups to prioritize them accordingly.

Alarms can notify users through the following methods:

- Alarm Indicator
- Alarm Viewer
- Alarm Distributor

TIP
For alarm optimization tips, see Optimize Alarm Usage.

13.1. Alarm Conditions

Alarms can have the following conditions:

Alarm condition	Description
Active	The alarm condition is met and remains unacknowledged.
Inactive	The alarm has returned to normal but is still unacknowledged.
Acknowledged	The alarm condition is met, but the alarm has been acknowledged.
Normal	The alarm has returned to normal and has been acknowledged.

Alarm triggering

- Alarm conditions are triggered by a fixed value.
- For analog alarm tags, the trigger value cannot be controlled from a register.
- Alarms cannot be set to trigger on intervals.

13.2. Alarm Indicator

The **alarm indicator notification window** remains visible in the iX App as long as there are active alarms, regardless of the current screen. By default, it appears in the upper-left corner but can be moved anywhere on the screen.

The indicator's appearance changes based on the severity and status of the most critical active alarm. Its color, which reflects the current alarm status, can be customized per alarm group. The indicator will disappear once all alarms are acknowledged and have returned to an inactive state.

To set alarm statuses that trigger the alarm indicator:

1. Go to the Alarm Server Configuration Page.
2. Click **Settings**, to open the Alarm Server Properties Window.
3. Navigate to > **General** > **Show Alarm Indicator** and select the desired alarm condition.

Alarm condition	Indication
Active	Flashing red
Inactive	Flashing green
Acknowledged	Flashing green

4. Click **OK**.

13.3. Alarm Items

An **alarm item** represents a predefined condition that triggers an alarm when specific criteria are met. It is used to monitor real-time data, detect abnormal conditions, and alert users when a threshold or event is exceeded. Alarm items are typically configured with conditions such as values, states, or trends, and they can be associated with different actions, such as visual indicators, sound alerts, or notifications.

13.3.1. Add Alarm Items

To define a new alarm item:

1. Go to **Project Explorer** > **Functions** > **Alarm Server** > **Alarm Items** ,
2. Click **Add**
3. Configure the alarm settings according to the Alarm Server Configuration Page chapter.

13.3.2. Import and Export Alarm Items

You can import and export alarms with the **Import** button on the Alarm Server Function Configuration Page. This process follows a similar procedure as described in Import and Export Tags.

13.4. Alarm Groups

Alarms can be divided into groups, such as to indicate alarm priority. By default, one alarm group is predefined. Each alarm group can have its own color attributes, which can be customized. In the Alarm Viewer, alarms can be sorted by group.

13.4.1. Add Alarm Groups

To define a new alarm group:

1. Go to **Project Explorer** > **Functions** > **Alarm Server** > **Alarm Groups** ,
2. Click **Add**
3. Configure the alarm settings according to the Alarm Server Configuration Page chapter.

NOTE
Alarm filters in the iX App will revert to default settings if the filtered alarm group is renamed, see Filter Alarms in the Alarm Viewer.

13.5. Configure Actions for Alarm Items and Alarm Groups

Actions can be configured for both individual alarms and alarm groups. You can set up one or more actions for specific alarm events.

1. Navigate to the Alarm Server Configuration Page.
2. *For alarm items*: Go to the **Alarm Items** tab.
   For alarm groups: Go to the **Alarm Groups** tab.
3. In the **Action** column for the desired alarm item, click **...**.
4. Select **Trigger** and **Action**.
   - For information on available action triggers, refer to Alarm Server Action Triggers.
   - For information on available actions, refer to Available Actions.
5. Click **OK**.

13.6. Alarm Viewer

The Alarm Viewer Object displays detailed alarm information, including alarm texts from the alarm server, and lists the most recent alarms organized by alarm group. It provides more comprehensive details compared to the Alarm Indicator.

The **default design** of the Alarm Viewer adapts to the size of the selected HMI panel target, ensuring optimal visibility and usability, especially on smaller panels.

The **Alarm Viewer** can also be displayed as a popup window. For more information, see Manage Alarms Popup Window.

13.6.1. Customize the Alarm Viewer

- Most of the customization is done by selecting the Alarm Viewer object and navigating to the **General** ribbon tab. For available settings, refer to Alarm Viewer Object.
- **Font and button width settings** are adjusted in the Property Grid Window.
- **Color** for the Alarm Viewer window are set in the Property Grid Window.

> **NOTE**
> Some of the color effects require that **Vista Style** is activated under **Extended** in the Property Grid Window.

- **Alarm status colors** are set in the Alarm Server Properties Window.

13.6.2. Manage Alarms Popup Window

The **Manage Alarms** popup window in the Alarm Viewer Object allows you to enable or disable alarms in the iX App.

Activate Manage Alarms:

1. Select the Alarm Viewer Object,
2. Go to **General** > **Buttons** and select **Manage Alarms**.

This adds the **Manage Alarms** button to the Alarm Viewer and activates the feature.

Manage Alarms features and restrictions:

- Scripts and actions are tied to the Alarm Viewer Object.
- Font, size, and color settings are inherited from the Alarm Viewer Object; individual settings for the **Manage Alarms** popup are not available.
- Filter and group functions can be used to refine alarms for enabling/disabling.
- The **Select** and **Enable/Disable** actions only apply to alarms matching the filter criteria.
- Multi-select functionality is only available on PC.
- Enable/disable functionality is unavailable if a server address is assigned for remote alarm handling.

13.6.3. How to Use the Alarm Viewer in the iX App

13.6.3.1. Acknowledge Alarms on the Alarm Viewer

You can acknowledge alarms on the Alarm Viewer in the iX App in the following ways:

Acknowledge a specific alarm:

- Right-click on the alarm line and select **Acknowledge**.
- Select the alarm line and press the **Acknowledge Selected** button.

Acknowledge multiple alarms:

- Press the **Acknowledge All** button to acknowledge all current alarms.
- Press the **Acknowledge Visible** button to acknowledge only visible alarms on the associated Alarm Viewer control.

Clear acknowledged alarms:

- Press the **Clear** button to remove inactive acknowledged alarms from the list.
- Press the **Clear Visible** button to remove visible inactive acknowledged alarms.

Automate acknowledgement:

- Use script actions to automate alarm acknowledgements. For more information on scripting, see the Scripts section.

13.6.3.2. Enable and Disable Alarms in the Alarm Viewer

To prevent the Alarm Server from processing unnecessary alarms, disable irrelevant alarms in the Alarm Viewer in the iX App.

Features and limitations

- Disabled alarms are listed with a timestamp in a disabled list.
- The main alarm page shows the count of disabled alarms below the alarm list.
- Alarm timestamps are preserved across reboots.
- Disabled alarms stay inactive until re-enabled in the Alarm Viewer.
- Multi-select functionality is only available on PC.
- Enable/disable functionality is unavailable if a server address is assigned for remote alarm handling.

13.6.3.3. Sort the Alarm List in the Alarm Viewer

To sort the alarm list in the iX App, press a column heading. This action temporarily pauses the Alarm Viewer.

13.6.3.4. Filter Alarms in the Alarm Viewer

Press the **Filter** button in the Alarm Viewer to open a dialog where you can create a filtering profile. This profile controls which alarms appear in the alarm list. The **Filter** button only displays panels that match the project's panel type.

You can filter alarms based on any combination of the following conditions:

- **Alarm group** – The first column in the filter window lists alarm groups. Select **Enable** to display alarms from a specific group.
- **Alarm status** – Displays alarms with statuses that match the selected values.

If you rename a filtered **Alarm Group**, the alarm filters will reset to default settings.

13.6.3.5. Pause the Alarm Viewer

Press the **Pause** button to pause the Alarm Viewer, preventing new alarms from appearing. This allows you to inspect and sort the alarm list without interruptions. The alarm list also pauses automatically when you select an alarm.

The alarm list resumes updating in the following scenarios:

- You press the Play button.
- You switch screens and then return to the same screen or another screen displaying the Alarm Viewer.
- A 5-minute timeout occurs.

13.6.3.6. Configure Actions in the Alarm Viewer

Press the **Info** button in the Alarm Viewer to choose an action from a list. Actions is applied in the following order:

1. **Individual alarm** – If the alarm has a specific action assigned, it is applied first.
2. **Alarm group** – If no action is assigned to the alarm, the action applies to its group.
3. **Alarm server** – If no action is specified for the group, it applies to the alarm server.

13.7. Alarm Distribution

This chapter outlines how to enable and configure internal alarm distribution, allowing devices to act as both clients and servers. It covers setting up alarm routes, receivers, filters, and schedulers, as well as configuring devices for alarm collection and distribution.

13.7.1. Enable Internal Alarm Distribution

To enable internal alarm distribution functions, which act as both a client and server, follow these steps:

1. Click **Insert** > **Functions** > **Alarm Distributor** to enable the Alarm distributor.
2. Go to the Alarm Server Configuration Page and click **Settings**.
3. Select **Alarm Distribution** and enable **Enable Sending To Alarm Distributor Server**.
4. Click **OK**.

5. Configure **Alarm Items** and **Alarm Groups** according to the Alarm Server Configuration Page chapter.
6. Go to the Alarm Distributor Configuration Page.
7. Add a **Route** and configure **Receivers**, **Alarm Filter** and **Scheduler** according to the Alarm Distributor Configuration Page chapter.
8. On the Alarm Distributor Configuration Page click **Configure Distribution Devices**.
9. Configure the device according to the Configure Distribution Devices Window chapter.

13.7.2. Set Up Alarm Collection for Distribution

To enable distributed alarm functions on another HMI panel or PC acting as a server, follow these steps:

1. Click **Insert** > **Functions** > **Alarm Distributor** to enable the Alarm distributor.
2. Go to the Alarm Distributor Configuration Page.
3. Add a **Route** and configure **Receivers**, **Alarm Filter** and **Scheduler** according to the Alarm Distributor Configuration Page chapter.
4. On the Alarm Distributor Configuration Page click **Configure Distribution Devices**.
5. Configure the device according to the Configure Distribution Devices Window chapter.

13.7.3. Modify Alarm Distribution Settings

To modify the alarm distribution settings in the iX App, use actions. For more information, see Configure Actions in the Alarm Viewer.

If you use the Alarm Distribution Viewer or **alarm distribution actions** in iX App, a notification appears when downloading the project to HMI panels. You can then choose to:

- Overwrite the panel settings.
- Retain the current settings.

14. Servers

Servers provide essential communication functions for accessing and managing data in iX Developer projects. This chapter covers the configuration and security aspects of the Web Server and OPC UA Server.

IMPORTANT
Security is a critical aspect when enabling servers. Setting strong passwords and following best practices for authentication and encryption are highly recommended to prevent unauthorized access.

14.1. Web Server

The **Web Server** enables secure data exchange through REST APIs using HTTPS, allowing tag management via API endpoints. It supports authentication, bearer tokens, and a range of security measures to protect data.

About the new Web Server in iX Developer 3:

- **Client JavaScript SDK deprecated** – The JavaScript SDK is no longer maintained and should not be used in new projects.
- **REST API compatibility** – API Endpoints have been updated and are **not** fully backward compatible with previous versions.
- **HTTPS-only communication** – The web server now supports only HTTPS for secure data transmission.
- **Fixed local port** – The local port is hardcoded to **7001** and cannot be changed.

14.1.1. Web Server Settings

To open the web server settings, go to **Insert** > **Server** > **Web Server**.

Parameter		Description
Enable static files		Enabling static files allows you to add your own files that the web server can access. This feature is useful for including web pages and images. Click **Open Files** to open the directory where you can add files and folders. Since the website root is located at **https://[panelIP]/[project name]/index.html**, you must use relative paths to embed stylesheets and other resources. For example, if your stylesheets are in the **Css** folder, use the following format: `<link rel="stylesheet" href="../Project28/css/site.css">` This example correctly links to the stylesheet.
API	**Enable swagger**	This option enables API documentation, which you can view as a web page on the web server. Navigate to https://localhost:7001/Swagger to access it.
	Enable read tags	This option allows you to read tags. The access level for individual tags must be set to **Read** or **ReadWrite**.

Parameter		Description
	Enable write tags	This option allows you to update tags. The access level for individual tags must be set to **Write** or **ReadWrite**.
Edit Credentials		Enter the username and password needed for authentication on the web server. This step is required to access the web server. **PASSWORD POLICY** Passwords must be at least 12 characters long and include an uppercase letter, a lowercase letter, a digit, and a special character from this set: `!@#$%^&*()_-+=[{]};:>

14.1.2. Example - Configure Web Server

VIDEO TUTORIAL

https://player.vimeo.com/video/1060518278?badge=0&autopause=0&player_id=0&app_id=58479

14.1.3. API Endpoints

This section lists the API endpoints available in the Web Server.

URL

- **Local Access**: https://localhost:7001
- **IP Access**: https://[panelIP]/[project name]

Security and Response format

- **Allowed SSL Protocols**: TLS 1.2, TLS 1.3
- **Response Data Format**: JSON

API documentation

- **Swagger**: https://localhost:7001/Swagger

TIP

To access the **Swagger** page, you must first enable it in iX Developer. For instructions, refer to the Swagger chapter.

14.1.3.1. Authenticate User

Method: POST /Authenticate
Authenticates the user and returns a bearer token and a refresh token.

Request body

Element	Description	Type	Required
`userName`	Username used for authentication	String	Required
`password`	Password used for authentication.	String	Required

Example request

```
{
    "userName": "Arne",
    "password": "kljhyihjhgty1!F"
}
```

Response

Element	Description	Type	Note
`token`	Bearer token to include in the header for subsequent calls.	String	Valid for **1 h**.
`refreshToken`	New refresh token to obtain a new bearer token using the Refresh Authentication Token endpoint.	String	Valid for **4 h**.

Example response

```
{
    "token": "ABC123",
    "refreshToken": "CDY123"
}
```

Status codes

Code	Description	Notes
200	OK	Authentication successful.
401	Unauthorized	Authentication failed.
429	Too many requests	Rate limit exceeded.

14.1.3.2. Refresh Authentication Token

Method: POST /Token/Refresh
Retrieves a new **bearer token** and **refresh token**.

Request body

Element	Description	Type	Required	Notes
`refreshToken`	Refresh token used for authentication.	String	Required	Becomes invalid after use and cannot be reused.

Example request

```
{
    "refreshToken": " CDY123"
}
```

Response

Element	Description	Type	Note
`token`	Bearer token to include in the header for subsequent calls.	String	Valid for **1 h**.
`refreshToken`	New refresh token to obtain a new bearer token using the Refresh Authentication Token endpoint.	String	

Example response

```
{
    "token": "POKJ34F",
    "refreshToken": "XCV34"
}
```

Status codes

Code	Description	Notes
200	OK	Token refreshed successfully.
401	Unauthorized	Authentication failed.

14.1.3.3. Revoke Refresh Token

Method: POST /Token/Revoke
Revokes (invalidates) the specified refresh token.

Request body

Element	Description	Type	Required
`refreshToken`	Refresh token to be revoked.	String	Required

Example request

```
{
    "refreshToken": " CDY123"
}
```

Status codes

Code	Description	Notes
200	OK	The refresh token was revoked.
401	Unauthorized	Authentication failed.

14.1.3.4. Retrieve All Tag Names

Method: GET /Tags
Retrieves a list of all tag names.

Headers

Header name	Description	Required
`Authorization`	Access token for authentication. Use this format: `Bearer {token}`. Replace `{token}` with a valid bearer token.	Required

Example request

```
GET https://10.101.100.100/Proj1/Tags
Authorization: Bearer ABC123
```

Response

Element	Description	Type
`{name}`	An array containing all tag names. `{name}` represents a tag name.	Array of strings

Example response

```
[
    "Tag1",
    "Tag2"
]
```

Status codes

Code	Description	Notes
200	OK	Request was successful.
401	Unauthorized	Authentication failed.
403	Forbidden	Access is denied.
500	Internal Server Error	An unexpected error occurred.

14.1.3.5. Retrieve Tag Data

Method: GET /Tags/{name}
Retrieves detailed information about a specified tag. Replace `{name}` with the tag name.

Headers

Header name	Description	Required
`Authorization`	Access token for authentication. Use this format: `Bearer {token}`. Replace `{token}` with a valid bearer token.	Required

Example request

```
GET https://10.101.100.100/Proj1/Tags/Tag1
Authorization: Bearer ABC123
```

Response

Element	Description	Type	Note
`name`	The tag name.	String	
`value`	The tag value.	String	
`dataType`	The tag's data type.	String	See Data Types.
`isReadOnly`	Indicates if the tag is read-only.	Boolean	
`quality`	The tag's quality status.	String	See Quality.

Example response

```
{
    "name": "Tag1",
    "value": "0",
    "dataType": "Int16",
    "isReadOnly": false,
    "quality": "Good"
}
```

Status codes

Code	Description	Notes
200	OK	Request was successful.
401	Unauthorized	Authentication failed.
403	Forbidden	Access is denied.
404	Not Found	The specified tag does not exist.
500	Internal Server Error	An unexpected error occurred.

14.1.3.6. Update a Tag Value

Method: PUT /Tags/{name}
Updates the specified tag with a new value. Replace `{name}` with the tag name.

Headers

Header name	Description	Required
`Authorization`	Access token for authentication. Use this format: `Bearer {token}`. Replace `{token}` with a valid bearer token.	Required

Request body

Element	Description	Type	Required
`{value}`	The value to set the tag to. Insert the new value in `{value}`.	String	Required

Example request

```
PUT https://10.101.100.100/Proj1/Tags/Tag1
Authorization: Bearer ABC123

“5”
```

Status codes

Code	Description	Notes
200	OK	The tag was updated successfully.
400	Bad Request	Invalid input.
401	Unauthorized	Authentication failed.
403	Forbidden	Access is denied.
404	Not Found	The specified tag does not exist.
500	Internal Server Error	An unexpected error occurred.

14.1.3.7. Retrieve Data for Multiple Tags

Method: GET /Tagsbatch
Retrieves information for the specified tags.

Query parameters

Parameter	Description	Type	Required
`name`	The tag name.	String	Required

NOTE
You can append as many tag names as needed to the query string.

Headers

Header name	Description	Required
`Authorization`	Access token for authentication. Use this format: `Bearer {token}`. Replace `{token}` with a valid bearer token.	Required

Example request

```
GET https://10.101.100.100/Proj1/TagsBatch?name=tag1&name=tag2
Authorization: Bearer ABC123
```

Response

Element		Description	Type	Note
`status`		The status of the response.	String	
`succeeded`		Lists the information of the requested tag.	Array of elements	
`failed`		Lists the tag names and the reasons for failed updates.	Array of elements	
	`name`	The tag name.	String	
	`value`	The tag value.	String	
	`dataType`	The tag's data type.	String	See Data Types.
	`isReadOnly`	Indicates if the tag is read-only.	Boolean	
	`quality`	The tag's quality status.	String	See Quality.
	`reason`	The reason for the failed request.	String	

Example response

```
{
    "status": "PartialSuccess",
    "succeeded": [
        {
            "name": "Tag1",
            "value": "0",
            "dataType": "Int16",
            "isReadOnly": false,
            "quality": "Good"
        }
    ],
    "failed": [
        {
            "name": "tag2",
            "reason": "NotFound"
        }
```

```
    ]
}
```

Status codes

Code	Description	Notes
200	OK	Information on all or some tags was retrieved. See response example above for when only some tags was retrieved.
400	Bad Request	Information for no tags was retrieved.
401	Unauthorized	Authentication failed.
403	Forbidden	Access is denied.
500	Internal Server Error	An unexpected error occurred.

14.1.3.8. Update Values for Multiple Tags

PUT /Tagsbatch
This method updates the values for multiple tags in a single request.

Request body

Element	Description	Type	Required
`name`	The tag name.	String	Required
`value`	The tag value.	String	Required

NOTE
You can include multiple comma-separated tags in the request body.

Example request

```
PUT https://10.101.100.100/Proj1/TagsBatch
Authorization: Bearer ABC123

[
    {
        "name": "Tag1",
        "value": "1"
    },
    {
        "name": "Tag4",
        "value": "1"
    }
]
```

Response

Element		Description	Type	Note
`status`		The status of the response.	String	
`failed`		Lists the tag names and the reasons for failed updates.	Array of elements	Only present for failed requests.
	`name`	The tag name.	String	Only present for failed requests.
	`reason`	The reason for the failed request.	String	Only present for failed requests.
`successCount`		The number of successfully updated tags.	Integer	

Example response

```
{
    "status": "PartialSuccess",
    "failed": [
        {
            "name": "Tag4",
            "reason": "NotFound"
        }
    ],
    "successCount": 1
}
```

Status codes

Code	Description	Notes
200	OK	All or some tags were successfully updated.
400	Bad Request	All tags fail to update
401	Unauthorized	Authentication failed.
403	Forbidden	Access is denied.
500	Internal Server Error	An unexpected error occurred.

14.1.3.9. Swagger

Swagger is a tool for documenting REST APIs based on the OpenAPI Specification. In iX Developer, the API documentation is available as a web page on the web server.

How to open the iX Developer Swagger page

1. **Enable Swagger** in the Web Server Settings.
2. **Run** the project on the development PC.
3. Open a web browser and navigate to https://localhost:7001/Swagger.

NOTE
If you see a warning regarding your connection being insecure, it is due to the use of a self-signed certificate. You can safely ignore the warning by clicking **Advanced** and selecting **Continue to localhost (unsafe)**.

Listed you'll find all available **API endpoints** and **schemas**.

14.1.3.10. Value Formatting Guidelines

When working with different data types, use the following formats for consistency:

Format	Guideline	Example
Floating point	Values should be represented as strings in standard floating-point format.	`"1.25"`
DateTime	Dates and times should follow the ISO 8601-1:2019 standard.	`"2024-04-18T07:43:47"`
Integer	Integer values should be provided as strings.	`"546565"`
Boolean	Boolean values should be represented as "true" or "false".	`"true"` or `"false"`

14.1.3.11. Data Types

Data type	Description
Int32	A signed 32-bit integer type with values between -2,147,483,648 and 2,147,483,647.
Int16	A signed 16-bit integer type with values between -32,768 and 32,767.
Single	A floating-point type representing values from approximately 1.5 x 10⁻⁴⁵to 3.4 x 10³⁸ with a precision of 7 digits.
Double	A floating-point type representing values from approximately 5.0 x 10⁻³²⁴to 1.7 x 10³⁰⁸ with a precision of 15-16 digits.
DateTime	A type representing a date and time value.
String	A type representing Unicode character strings.
Boolean	A type representing Boolean values, either true or false.
UInt16	An unsigned 16-bit integer type with values between 0 and 65,535.
UInt32	An unsigned 32-bit integer type with values between 0 and 4,294,967,295.

14.1.3.12. Quality

The quality can be one of the following values:

- **Good**
- **Bad**
- **Unknown**

14.2. OPC UA Server

The **OPC UA Server** enables real-time data exchange between the HMI panel and controllers. It supports structured tag management, authentication, and namespace configuration, ensuring reliable integration with industrial systems.

TIP

For security guidelines on OPC UA servers, see OPC UA Security.

NOTE

OPC DA (Data Access) is no longer supported. When you open an existing OPC DA project, the system notifies you and provides an option to export invalid OPC DA tags.

14.2.1. OPC UA Server Overview

14.2.1.1. OPC UA Server Certificate

- A self-signed certificate is auto-generated when the server starts for the first time, valid for 20 years.
- Certificates exclude IP addresses, accommodating potential IP changes.

14.2.1.2. Supported OPC UA Features

- **Data Access (UA Part 8)** is supported.
- Reading and writing **objects, structures, and arrays** are not supported.

14.2.1.3. Address space

- All non-array tags defined in the project are displayed under a folder named **Tags** in the **Namespace TagProvider**.
- Tags are published with scaling applied (offset and gain).
- All tags are represented as **Variable** in the NodeClass.

14.2.1.4. OPC UA Server Limitations

Limitation	Description
Discovery	Not supported.
Encoding / Transport	Supports only UA TCP Binary.
User authentication	Only username/password authentication. Single account only.
Security	No encryption support.
Sessions	Maximum of 20 simultaneous sessions. Sessions time out after 20 minutes of inactivity (timer resets on server restart).

Limitation	Description
Address input in GUI	Namespace and BrowseName must have unique combinations. Non-unique combinations cause client-side identification issues. NodeId Identifier can be used as an alternative.
NodeId Identifier type	Only Numeric and String NodeId types are supported. Unsupported types (e.g., Guid, Opaque) are excluded during import.
Tags with dynamic data types	Dynamic data types are not supported, which may lead to unexpected runtime values.

14.2.2. OPC UA Server Properties

The **OPC UA Server Properties** are accessible via **Insert** > **Servers** > **OPC UA Server**.

Available properties

Parameter		Description
Enable OPC UA Server		Activates the OPC UA Server functionality.
Port		Specifies the server's port number.
Allow anonymous login		Enables anonymous access. To require a secured login, uncheck this option and configure a username and password.
Expose tag names as OPC UA node identifier		Publishes tags using their names as the node identifier instead of a unique number. **IMPORTANT** If you change this setting or modify tag names, you must update all previous OPC UA integrations. Ensure that all clients with tag integrations are updated to reflect changes on the server.
Tag visibility	**All tags visible**	Makes all tags visible in OPC UA.
	No tags visible	Makes no tags visible in OPC UA.
	Customized	Allows tag visibility to be managed individually in the Tags Configuration Page.
Tag default visibility *(only available when **Tag visibility** is set to **Customized**)*		Defines the default visibility for newly created tags.
Allow deprecated security policies		Enables deprecated security policies for backward compatibility. If checked and username/password authentication is used, the deprecated policy **Basic256** will be applied instead of more secure policies.

14.2.3. Controller Settings for the OPC UA Server

To open the **controller settings** for the OPC UA server:

1. Go to the **Controllers** tab of the Tags Configuration Page,

2. Select the OPC UA server in list and click **Settings**.

Available settings

Parameter		Description
URL		Sets the URL to the OPC UA server. Only **UA TCP Binary** is supported, so the URL must start with `opc.tcp://`.
Authentication settings		Enables authentication. Choose between anonymous login or providing a username and password for secure access. **IMPORTANT** When using username/password, you can also enable **Allow non-secure connections** (for older servers such as 2.40 SP4 or earlier). Use this option cautiously!
Namespace settings		Defines a unique identifier for the namespace used by the OPC UA server. And assign a shorthand prefix to the namespace for easier reference in tag configurations.
Tag View settings	**Namespace separator character**	Defines the character separating the **namespace URI prefix** from **tag IDs**.
	Default namespace	Chooses the default **namespace URI prefix** for tag entry.
Subscription settings*	**Max subscriptions**	Sets the maximum subscriptions supported by the server (refer to the server manual).
	Max items per subscription	Sets the maximum number of items per subscription (refer to the server manual).
	Connection delay (in ms)	Defines the delay before the application connects to the server after starting. This option creates staggered connections between clients so not all clients attempt to connect simultaneously to the server.
Sampling settings	**Use poll group values**	Enables the option to base sampling intervals on poll group values.**
	Sampling interval	Sets the rate at which the server checks for data changes.**
	Only get the latest sampling interval when polling	If the sampling interval is faster than the publishing interval, a queue of values may form in the server. Enables to limit the queue to the latest value; if disabled, all samples are included in notifications.

* Parameters allow for performance optimization, but behavior may vary between servers.

** **Sampling intervals** depend on the server's **MinSupportedSampleRate** and **AvailableSamplingRates**. The final interval will not be lower than the minimum supported rate, and the closest available sampling rate to the desired value will be used.

14.2.4. Add and Enable the OPC UA Server

To add an **OPC UA Server** to your project, choose one of the following options:

- ***When creating a new project***: In the Choose Controller section of The Wizard, select **OPC UA Server**.
- ***For an existing project***: In the Controllers tab of the Tags Configuration Page, click **Add**, then select **OPC UA Server**.

Then, follow these steps:

1. Enter the **URL** to the OPC UA server and click **OK**.
2. Go to **Insert** > **Servers** > **OPC UA Server** and select **Enable OPC UA Server**.
3. Configure the settings as described in OPC UA Server Properties.

14.2.5. Add Tags from OPC UA Server

You can add tags from the OPC UA Server using three different methods. Below are the steps for each option:

NOTE

- **Array tags** are not supported by OPC UA controllers.
- The **Sync Namespace** button can be used to import or synchronize namespace settings without importing any tags.

14.2.5.1. Option 1: Manual Entry with Namespace URI Prefix and BrowseName

1. Go to the Tags Configuration Page.
2. Click **Add** and enter a **Name** for the added tag.
3. In the **Controllers** > **{OPC UA server name}** column, enter the following:
   `{namespace URI prefix}:{BrowseName}`
   - `Namespace URI prefix` is defined in the Controller Settings for the OPC UA Server.
   - If no `namespace URI prefix` is defined, the default namespace URI prefix from the controller settings will be used.

Example

```
NS3:tag
```

14.2.5.2. Option 2: Enter NodeId Information

1. Go to the Tags Configuration Page.
2. Click **Add** and enter a **Name** for the added tag.
3. In the **Controllers** > **{OPC UA server name}** column, enter the following:
   `*{namespace URI prefix}:{NodeId IdentifierType}:{NodeId identifier}`
   - `Namespace URI prefix` is defined in the Controller Settings for the OPC UA Server.

Examples

```
*NS3:Numeric:42949672
```

```
*NS3:String:Tag1
```

14.2.5.3. Option 3: Select Tags from OPC UA Server

1. Go to the Tags Configuration Page.
2. Click the down arrow next to the **Add** button and select **Add Tags from OPC Server [{OPC UA server name}]**.
3. In the **Browse OPC Server** dialog, select the tags you want and click **OK**.

14.2.5.4. Map Tag Properties to OPC UA Server Attributes

Tag property	UA server attribute
Name	`DisplayName` / `BrowseName`
Data type	`DataType`
Access right	`AccessLevel` / `UserAccessLevel`
Description	`Description`
Poll group interval	`MinimumSamplingInterval`

NOTE
The **BIT** data type is not supported by OPC UA. Tags of this type will be handled as **INT16** in the server. For best results, use the **BOOL** type instead.

14.2.6. Import Structured Tags from OPC UA Server

Structured tags can be imported and managed similarly to regular OPC UA tags. Depending on the selected elements in the browse dialog, the tags may either appear under the **Structured Tags** tab or as a flat list under the **Tags** tab in the Tags Configuration Page.

Prerequisites

- Ensure the Output Window is enabled before starting the import process for visibility into potential problems.

Instructions

1. Go to the Tags Configuration Page.
2. Click the down arrow next to the **Add** button and select **Add Tags from OPC Server [{OPC UA server name}]**.
3. Browse the **OPC Server** dialog, select a tree structure or leaf nodes and click **OK**.
4. Information about skipped nodes or issues encountered during the import process will be shown in the **Output** window.

TIP
Imported structured tags can be accessed and filtered from tag lists just like regular tags.

14.2.6.1. Synchronize Structured Tags

In dynamic projects, server nodes may undergo changes such as renaming, addition, relocation, or removal. The synchronization functionality ensures that the structured tags in the project are aligned with the server's updated content.

Key details

- Synchronization only updates previously imported structured tags.
- Changes to object-connected tags (e.g., renaming) during synchronization will break the connections, as the references will not be automatically updated.
- Only already imported structured tags are updated with this functionality.
- If tags connected to objects are renamed through synchronization, those tag connections will be lost (the name of the tag reference will not be updated).

15. Databases

A **database** in iX Developer is a structured collection of data organized to facilitate efficient storage, retrieval, and management. It is used within the iX Developer environment to manage data related to HMI applications, including alarms, recipes, and trends. The database structure enables the user to store and access data sets, which can be linked to specific Objects or used for logging purposes. Databases are crucial for real-time data management and help in creating dynamic and interactive user interfaces for industrial automation applications.

iX Developer uses **SQLite** as database. The database contents can be managed using third-party tools, and iX Developer provides a built-in Database Viewer Object to display data in the iX App.

15.1. Database Overview

The system uses three database files to store historical data, logs, and tag values.

Database	File name	Description
Database	`Database.db`	Stores data for: • Alarms • Data Loggers • Recipes
Audit Trail	`AuditTrail.db`	Stores Audit Trail events.
Non-volatile database	`NonVolatileDatabase.db`	Stores the values for **Non-volatile tags**, see Available Tag Types.

15.2. Change Location for the Active Databases

By default, the active database saves to the **SD card**, but you can change it to the internal **hard drive**. Before you switch, note:

- **Save to SD card** – Beijer Electronics recommend saving the database to an **SD card** to reduce flash wear on the internal drive, see Use SD Card to Reduce Flash Wear.
- **Database transfer** – When switching the database location, the system automatically copies the database to the new location during project transfer. This may **slightly delay the iX App startup**.
- **Script compatibility** – Changing the database location may affect existing scripts. The system **does not display build errors or warnings** for script issues caused by this change.

Instructions

1. Click **Project** > **Project** > **Settings** to open the Project Properties.
2. Select Database and choose **Database location**.
3. Click **OK**.

15.3. Fetch Databases From the HMI Panel

With the **Fetch Databases** command in the Project Ribbon Tab you can fetch the database from an HMI panel in the network environment to the development PC.

NOTE
The database can only be fetched from a panel running the same project (i.e. a project with the same name and of the same size) as the current iX Developer project.

Instructions

1. Connect the HMI panel to the development PC using an Ethernet connection.
2. in iX Developer, click **Project** > **Transfer** > **Fetch Databases** to open the **Fetch Databases** dialog.
3. The **Fetch Databases** dialog displays a list of connected panels with their IP addresses and types. Select the correct panel from the list.
4. Under **Project Name**, select project and click **Fetch**.
5. Eneter the login credentials for the **OS3 Administrator user account** and click **Authenticate**. For more information on OS3 users, see User Management In the **OS3 User Guide**.
6. Select location where to save the database and click **OK**. The panel will pause during the fetch, and will be restarted automatically afterwards.

TIP
To fetch databases without using iX Developer, use the **Transfer Client CLI**. For instructions, see Fetch Files from the HMI panel Using Transfer Client CLI.

15.4. Export Database From the iX App

Certain databases in your iX App can be exported as a `.csv` file by using an action. For details on how actions work, see the Actions chapter. The database can be saved to a USB drive, SD card, or the project folder on the HMI panel.

Databases available for export using these instructions:

- Alarm Server database.
- Audit Trail database.

If you want to export a Recipe database, see Export Recipe Database From the iX App.

NOTE
You cannot export databases during project simulation.

Instructions

1. In iX Developer, select the object that will trigger the action, then go to the Actions Ribbon Tab.
2. In the desired action trigger group, select **Select Action** > **Database** > **Database Export**.

NOTE
If exporting multiple databases, configure one action for each database.

3. Click the **...** button to open the **Database Export** dialog and set the export parameters.

   Export parameters

Parameter		Description
Select a database to export		Select which database to export.
Name of exported file		Select the name for the exported file. If the **Overwrite** box is unchecked, the export file name will include the date and time to prevent overwriting existing files.
Select the delimiter in the export data		Select the delimiter in the **.csv** file: **Comma** or **Semicolon**.
Select what happens if the files exceed the folder limit	**Delete older files**	Deletes the first 10% of files ordered by date once the folder limit is reached.
	Group files by date	Files will be organized into folders named with the current date. If a folder exceeds the file limit, a new folder will be created with the current date and a number suffix (e.g., 20170309, 20170309_1, 20170309_2).
File encoding		Select file encoding: **Unicode (UTF-8)**, **Unicode** or **Unicode (Big-Endian)**.

4. Click **OK**.
5. In the lowest dropdown, select export location:
   - **USB**: Saves to a USB flash drive.
   - **SD Card**: Saves to an SD Card.
   - **Select at Runtime**: Allows the user to choose the save location in the iX App.
   - **AppData**: Saves to `ProgramData\OS3 Packages\<<ProjectGUID>>\Exports\DatabaseExport`.

The object is now configured to export the database when triggered.

NOTE

- If **Enable FTP Friendly Names** is enabled in the Advanced section of the **Project Properties**, spaces are replaced with underscores, and periods (except for before the extension) are replaced with dashes.
- To display certain characters (such as Chinese or Arabic) correctly in Excel, select **Unicode (UTF-8)** in **File encoding**.

15.5. Back Up Databases in the iX App

The **Back Up Databases** action creates duplicate copies of the selected databases in the selected location. For more information on actions, see the Actions chapter.

Instructions

1. In iX Developer, select the object that will trigger the action, then go to the Actions Ribbon Tab.
2. In the desired Action Trigger group, select **Select Action** > **Database** > **Back Up Databases**.

3. Click **Configure** to open the **Back Up Databases** dialog.
4. Select the databases to back up:
   - **Database**: Backs up the `Database.db` file.
   - **Audit trail**: Backs up the `AuditTrail.db` file.
   - **Non-volatile database**: Backs up the `NonVolatileDatabase.db` file.

   TIP
   For information on the database files, see Database Overview.

5. Select backup destination:
   - **USB memory**: Saves to a USB flash drive.
   - **SD card**: Saves to an SD card.
   - **SD card for project restore**: Saves to an SD card for restoring a project. For details on restoring a database, see Restore Databases in the iX App.
   - **Select at Runtime (SD card for project restore not supported)**: Allow the user to choose the save location in the iX App. The user cannot select **SD card for project restore**.
   - **AppData for corrupt database recovery**: Saves to `ProgramData\OS3 Packages\<<ProjectGUID>>`.
6. Click **OK**.

The object is now set up to back up the selected databases when the action is triggered. When triggered in the iX App, a message will confirm that the databases are being backed up.

15.6. Restore Databases in the iX App

A database backed up with **SD for project restore** enabled (see Back Up Databases in the iX App) can be restored using the **Restore Databases** action. For more information on actions, see the Actions chapter.

Instructions

1. In iX Developer, select the object that will trigger the action, then go to the Actions Ribbon Tab.
2. In the appropriate Action Trigger group, choose **Select Action** > **Database** > **Restore Databases**.
3. Click **Configure** to open the **Restore Databases Configuration** dialog.
4. Select the databases to restore:
   - **Database**: Alarm, data logger, recipe.
   - **Non-volatile database**: Values for non-volatile tags.

   TIP
   For information on the database files, see Database Overview.

5. Click **OK**.
6. In the iX App, trigger the action.

7. When the **File Selector** opens. Select storage location and the database to restore, and click **OK**.
8. Click **OK** again to confirm the restore.

Restore behavior

- After restart, the system performs project validation and database schema comparison.
- **If validation fails**: The restore is canceled and the iX App starts without changes.
- **If validation is successful**: The selected database replaces the existing one, and the iX App starts with the restored data.

15.7. Run Database Cleanup From the iX App

Use the **Database Cleanup** action to optimize database size and performance. This action is helpful in the following situations:

Reclaim space	After deleting large amounts of data, unused space may remain in the database. Running a cleanup reduces the file size by reclaiming this space.
Improve performance	Frequent database activity (insert, update, delete) can cause fragmentation. Cleanup reorganizes tables and indexes for faster access.
Before fetching the project from the HMI panel	To reduce project size, first run the **Clear Datalogger** action, then run the **Database Cleanup** action.

TIP
For more information on actions, see the Actions chapter.

Instructions

1. In iX Developer, select the object that will trigger the action, then go to the Actions Ribbon Tab.
2. In the desired Action Trigger group, choose **Select Action** > **Database** > **Databases Cleanup**.
3. To clean up the database, trigger the object in the iX App.

15.7.1. What Happens During Cleanup

The system creates a temporary copy of the database, then replaces the original file with the cleaned version. This process uses a rollback journal or write-ahead log (WAL), which may require free disk space up to twice the size of the database.

NOTE
The action will fail if there are open transactions or active SQL statements when it starts.

16. Tags

This section explains how to work with **Tags**. Tags allow objects to interact with a controller by changing their values. These changes can affect the appearance of objects on screen. Until connected to a tag, screen objects remain static.

Tags are managed from the Tags Configuration Page. Each tag has a symbolic name and a data type. Tags can be associated with a connected controller, be internal, or belong to the system. A tag can also be connected to multiple controllers simultaneously.

TIP
For guidelines on to improve tag performance, see the Tag Handling chapter.

16.1. Available Tag Types

Tags in iX Developer are organized by function and source into several main types. Each tag type has unique properties and behaviors based on its role in the project.

Controller Tags (External Tags)

Connect to external controllers or PLCs. These tags allow data to be read from or written to the controller.

Internal Tags

Used for calculations and local values that do not need to be stored in the controller, typically for operator-only displays. Any number of volatile, user-defined internal tags can be created. The order of internal tags does not affect functionality.
Internal tags can be:

- **Volatile** - values are not saved after a project restart.
- **Non-volatile** - values are retained if this option is enabled.

These settings are specified in the Tags Settings. By default, internal tag values are not saved on the HMI panel after a restart unless **Non Volatile** is selected.

System Tags

Used to access system variables such as *Used RAM* or *DateTime*. These tags appear in blue in the tag list. System tags can be used in screens, logging, or scripts. For example, **Date and Time** tags access the controller clock in the panel.

NOTE
Do not set system tags as **Non Volatile** in the Tags Settings, as this may cause flash memory wear.

For a list of all available system tags, see Available System Tags.

User-Defined Tags

Custom tags created for specific project requirements. These may be internal or external.

Array Tags

Used to store multiple values in a single tag. Supported only by controllers that allow incremented addressing.
For setup instructions, see Set up Array Tags.

16.1.1. Available System Tags

System tag group	System tag	Description
Communication	**Communication Error Message**	Latest communication error message.
	Communication Errors	Number of active communication errors on all controllers.
	Remote Alarm Server Connection Errors	Number of remote alarm servers with connection issues, e.g. disconnected servers.
Database	**Database Error Active**	Set to `1` when there is an active database error.
	Database Error Message	Latest database error message.
	Database Max Size Exceeded	Set to `1` when the database maximum size is exceeded.
Date and Time	**DateTime**	Current date and time.
	Day / Day of Week / Month / Year	Individual date components (1 = Sunday).
	Hour / Minute / Second	Individual time components.
Project	**Current Screen Id** (`int16`, *read-only*)	ID of the currently displayed screen. Value `-1` if no ID is set. This value updates each time a new screen is opened, except when opening popup screens. `0` is also a valid screen ID.
	Current Screen Name	Name of the current screen.
	Current User	Logged-in user name.
	Debug Logging Enabled	Indicates if debug logging is active.
	Latest Loaded Recipe	Name of the most recently loaded recipe.
	Max Screen Cache Memory Load	Memory used by screen cache (% of RAM).
	New Screen Id (`int16`, *read/write*, *always active*)	Set this tag to trigger a screen change. When the value of the New Screen ID tag changes, the system navigates to the screen with the corresponding ID. If the specified screen ID does not exist, a notification message is displayed. If the new screen has configured Instances, the default instance will be displayed.
	Number of Disabled Alarms	Total number of disabled alarms.
System	**Available RAM / Available Storage**	Free memory available (RAM in kB, storage in MB).

System tag group	System tag	Description
	Backlight Brightness Level	Backlight brightness level (%). **NOTE** Brightness changes close to 0% may not be noticeable. Full dimming (0%) is available only on **Marine** HMI models.
	CPU Load / CPU Load Core X	Total and per-core CPU usage (%).
	OS Name / OS Version	Operating system name and version.
	Physical RAM / Storage Memory	Installed RAM and storage (in kB/MB).
	Used RAM / Storage	Used RAM and storage (in kB/MB).
	Used RAM Percent / Used Storage Percent	Used RAM and storage (in %).

16.2. Add Tags

To add tags to the project:

1. In the Project Explorer Window, click **Tags** to go to the Tags Configuration Page.
2. Click **Add** to create a new tag.
3. Set the desired tag settings (see the Tags Configuration Page). If predefined values are available, you can select them by clicking the arrow on the right side of the table cell.

To add tags to an object:

1. Select the object.
2. Go to **Home** > **Tag/Security** or **General** > **Tag/Security** and **Select Tag**.

Tags added to an object remain **internal** (see Available Tag Types) until they are connected to a controller on the Tags Configuration Page.

16.3. Remove Unused Tags

The **Remove Unused Tags** dialog allows you to detect and remove tags that are not in use. To open it:

1. In the Tags Configuration Page, go to the **Tags** tab.
2. Click the arrow next to the delete button and select **Remove Unused Tags**.

IMPORTANT
Carefully review all tags listed for removal. Deleting tags that are still in use, especially those in scripts, may corrupt the project. Confirm that tags are inactive before removing them to ensure project integrity. For more details on using tags in scripts, refer to the Tags in Scripts chapter.

16.3.1. Automatic Detection of Unused Tags During Build

When building the project, unused tags are automatically detected, and the **Remove Unused Tags** dialog will appear. To enable or disable this automatic detection:

1. Go to **File** > **Options** > **Build options**.
2. Set **CheckForUnusedTags** to **True** (to enable) or **False** (to disable).

16.4. Import and Export Tags

iX Developer allows importing and exporting tags to facilitate editing in external tools like *Excel* or text editors, or for transferring tags between projects. Both the entire taglist and specific tags related to connected controllers can be managed.

16.4.1. Import Tags

Instructions

1. In the Tags Configuration Page, go to the **Tags** tab.
2. Click on the arrow next to the **Import** button.
3. Select either **Import Complete Taglist** or **Import Tags to {Controller}**. This will open the **Import Tags** dialog.
4. Choose the **Import module** file format: **Text file** or **Excel file**.
5. Under **Filename**, select the file to import.
6. Set the **Column Separator** and specify **Import start at row**. Click **OK**.
7. Ensure tag definitions are correct and controller references are updated.

16.4.1.1. Save the Import Configuration

Instructions

1. Configure the import settings as described above.
2. Click **Save mapping as import module** before import.
3. The saved configuration will appear in the **Import module** dropdown next time the **Import Tags** dialog is opened.

16.4.1.2. Import Tags via the Command Prompt

You can add tags to the current iX Developer project using the **Command Prompt**. Follow these steps:

1. Open the **Command Prompt**.
2. Enter the following command:

```
{Executable file} “{Project location}\{Project name}.neoproj” /
ImportTags=”{Tag file location}\{Tag file}”
```

Replace the placeholders with the correct file paths.

3. If the import is successful, no application window will be displayed.

TIP

You can combine the `/ImportTags` switch with the `/RemoveTagsNotInFile` switch. This removes all existing tags in the controller that are not included in the import file.

Example

To open the project `Test1`, and import tags from the file `VarFile.txt` located in `C:\MyProjects`, use this command:

```
NeoIDE.exe “C:\MyProjects\Test1\Test1.neoproj” /
ImportTags=“C:\MyProjects\VarFile.txt”
```

16.4.2. Export Tags

Instructions

1. In the Tags Configuration Page, go to the **Tags** tab.
2. Click on the arrow next to the **Import** button.
3. Choose either **Export Complete Taglist** or **Export Tags to [{Controller}]**. This will open the **Export Dialog**.
4. Select the export **Format**: **Text file** or **Excel file**.
5. Enter a **Filename** and browse to the save location.
6. Define the **Column Separator** and select which columns to export.
7. Use the arrow buttons to adjust column order, if needed.
8. Click **Export**.

16.4.3. Update and Import Tags After Controller Change

If the controller is replaced with one that uses different tag names, follow these steps:

Export existing tags

1. In the Tags Configuration Page, go to the **Tags** tab.
2. Click on the arrow next to **Import**, then select **Export tags from [{controller}]**.
3. Set **Format** to **Text file**.
4. Enter a **Filename** and browse to the save location.
5. Define the **Column Separator** and select which columns to export.
6. Use the arrow buttons to adjust column order, if needed.
7. Click **Export**.

Modify tags in text file

1. Open the exported file in a text editor or spreadsheet application.
2. Update all existing tags to match those used by the new controller.
3. Save the file.

```
// Name,DataType,Size,AccessRight,Address_1,AccessRight_1 //
RecTag1,INT16,1,ReadWrite,40010,None
TankLevel,INT16,1,ReadWrite,40000,None
Valve1,BIT,1,ReadWrite,00000,None
Valve2,BIT,1,ReadWrite,00001,None
IsRunning,BIT,1,ReadWrite,00002,None
RecTag2,INT16,1,ReadWrite,40011,None
RecTag3,INT16,1,ReadWrite,40012,None
Tag4,DEFAULT,1,ReadWrite,,
```

Import updated tags

1. In the Tags Configuration Page, go to the **Tags** tab.
2. Click on the arrow next to **Import**, then select **Import tags from [{controller}]**.
3. Select the **Import module**, browse to the updated **Filename**, and define the **Column separator** and **Import start at row**.
4. Click **Import**.

To reuse this configuration later, see **Save the Import Configuration** in the Import Tags chapter.

16.4.3.1. Name Conflicts During Import

During import, you can resolve tag name conflicts in the following ways:

- **Rename** – Assign a new name to the imported tag.
- **Overwrite** – Replace all properties of the existing tag.
- **Merge** – Retain existing data where imported data is empty.
- **Skip** – Omit the conflicting tag.

All actions are recorded in a log file.

Examples

- **Tag1** with address **C1**, merged with a file where **Tag1** has no address, results in **Tag1** keeping address **C1**.
- **Tag2** with no address, merged with a file where **Tag2** has address **C2**, results in **Tag2** receiving address **C2**.

16.5. Array Tags

An array tag is a tag that holds multiple values.

If the **Array Size** is > 1, the tag is treated as an array tag.

16.5.1. Set up Array Tags

Prerequisites

- A controller with addresses that can be incremented numerically. See the controller's Driver Manual.

> **NOTE**
> Not all communication drivers have support for array tags.

Instructions

1. Add the tag as described in the Add Tags chapter.
2. On the Tags Configuration Page, go to the **Tags** tab.
3. On the desired tag in the tags list, go to **Controllers** > **Data Type** column and click the down-arrow next to **Data Type**.
4. Set the **Data Type**, **Size**, and **Array Size** values.
   If the **Array Size** is > `1`, the tag is treated as an array tag.

> **ARRAY TAG LIMITATIONS**
> - When using controllers with named variables and accessing individual bits with the syntax `VariableName.BitNo`, array tags of data type `BOOL` with bit addressing are only supported for **16-bits** tags.

16.6. Triggers

Triggers control when data exchange occurs within iX Developer. They help ensure data is exchanged at the right time based on the needs of the project.

There are two types of triggers:

- **Tag-based trigger** – Starts data exchange when a specific tag value changes. This supports real-time updates when a condition is met.
- **Time-based trigger** – Starts data exchange at regular time intervals. This is useful for periodic updates, even if no tag values change.

Configure triggers from the **Triggers** tab on the Tags Configuration Page.

16.7. Poll Groups

Poll Groups let you organize tags to be polled at five different time intervals.

16.7.1. Assign a Poll Group to a Tag

Instructions

1. In the Tags Configuration Page, go to the **Poll Groups** tab.
2. Enter a **Name** and set the **Interval** in milliseconds (ms).
3. Go to the **Tags** tab and select the desiredpoll group from the **Poll Groups** column.

TIP
If the **Poll Groups** column is not visible, enable it by selecting **Columns Visible** > **Others**.

16.8. Index Registers

Index Registers allow an operator to choose which tag an object uses to display its value. Unlike fixed addressing, where an object is always linked to the same controller address, index registers enable more flexible addressing.

16.8.1. Link Index Registers to Tags

Instructions

1. In the Tags Configuration Page, go to the **Index Registers** tab.
2. Enter a **Name** and select a **Controller**. You can use up to eight index registers per controller, and each register can be assigned to multiple objects.
3. Go to the **Tags** tab and select the index register from the **Index Register** column. Index registers can be linked to any tag in the global tag list, but each tag must be assigned to the correct controller.

TIP
If the **Index Register** column is not visible, enable it by selecting **Columns Visible** > **Others**.

16.8.2. Index Addressing Formula

The formula for index addressing is:

```
Display Value = Content in Tag (Object's Address + Content in Index ▶
Register)
```

Addressing example

If the index register contains the value **2** and the tag address specified for the object is **100**, the object will display the value from address **102**. If the index register value is changed to **3**, the object will display the value from address **103**.

16.9. Scaling

Scaling adjusts the display scale of controller values using the following formula:

Panel value = Offset + (Gain × Register value)

For values set from the **HMI panel**, scaling uses this formula:

Register value = (Panel value - Offset) / Gain

Scaling only affects controller-connected tags and does not change the defined maximum or minimum values, nor does it alter the number of decimal places.

16.9.1. Set Up Scaling for Controller Values

Instructions

1. In the Tags Configuration Page, go to the **Tags** tab.
2. Enable **Columns Visible** > **Scaling**.
3. In the **Scaling** column, configure the scaling parameters according to the formulas in the Scaling chapter. For information on **Read and Write Expression**, see Configure Read and Write Expressions for Tags.

16.10. Expressions

Expressions modify a tag's value before applying it to an object. They allow operations such as extracting a specific bit, adding a dynamic offset, or applying a gain. The modified value is then used by the object.

An expression is a single-line return statement written in C#. The default expression is `value`.

TIP
You can assign the same expression to multiple objects in a project.

16.10.1. Add and Configure Expressions for Objects

Expressions added to an object are available only within the current project. If the object is saved to the Component Library Window, it can be reused in other projects, but the expression itself will not be included. To reuse expressions across multiple projects, save them in the Manage the Expressions Library.

To add and configure expressions

1. Right-click the desired object. In the Mini Toolbar, click **...** next to **Select Tag** to open the **Expressions** dialog.
2. *To add an existing expression to the object:* Select it from the **Select Expression** dropdown menu.
   To add a new expression to the object: Click **New** and choose tags, operators, and functions from the available options to build your expression. You can also manually type the expression. If the expression is invalid, an error message will appear during project validation.

   TIP
   IntelliSense (auto-completion) will not assist when manually typing expressions

3. Click **OK**. The expression will run whenever the value of the original tag connected to the object changes.
4. To define **Offset** and **Gain**, see Scaling.

16.10.1.1. Manage the Expressions Library

The **Expressions Library** allows you to manage and reuse expressions across different projects.

To manage the expressions library

1. Right-click on an object. In the Mini Toolbar, click **...** next to **Select Tag** to open the **Expressions** dialog.
2. Under **Library Expressions**, click **Load**, **Save** or **Delete**.

NOTE
Pre-installed expressions, such as how to extract bits from an integer, are included with the installation of iX Developer. These expressions cannot be deleted

16.10.2. Expression Limitations

When working with expressions in iX Developer, keep the following limitations in mind:

- An expression is executed only when the value is *read*, i.e. when the value changes in the controller.
- Expressions are **not** executed when writing a value (e.g., from an analog numeric tag).
- If the expression references a tag other than the original tag, it will not be triggered by changes in the other tag's value.
- If an expression used in multiple locations is renamed or deleted, the references will not update automatically, leading to a build error.

Expressions cannot be used for the following data connections

- Actions
- Property grid
- Recipe item tags
- Current language tag in multiple languages configuration
- Trigger tags
- Index tags
- Scheduler tags
- Dynamic text for alarm items

Boolean comparisons

Expressions require explicit boolean comparisons.

- **Avoid**: `value`; `Globals.Tags.Tag1.Value`
- **Use**: `value != 0`; `Globals.Tags.Tag1.Value == 1`

16.10.3. Configure Read and Write Expressions for Tags

Read and Write Expressions allow you to manipulate tag values before reading them from or writing them to a controller. There are two types of tag expressions:

Read Expressions	Modify the tag value when the controller updates it.

Write Expressions	Modify the tag value when the user interface or a script updates it.

TIP

If these columns aren't visible in the **Tags** tab of the Tags Configuration Page, enable **Columns Visible** > **Scaling**.

NOTE

- Use **Read and Write Expressions** only for tags linked to a controller address.
- If a tag has both read and write expressions, ensure they are inverse operations to maintain data consistency.
- When using a write expression, make sure the controller's data type can accommodate the result. For example, if multiplying an integer by 0.1, set the controller's data type to **float**.

16.11. Data Exchange

iX Developer supports data exchange between controllers, even when using different brands. This exchange can be triggered by a digital tag or set to run at specific intervals. Data exchange can be configured for individual tags, array tags, or entire tag ranges. Importantly, this feature operates independently from general tag/controller settings; for instance, a tag that is read-only for an operator can still be read and written via data exchange.

16.11.1. Set Up Data Exchange

Instructions

1. In the Tags Configuration Page, go to the **Tags** tab.
2. Enable **Columns Visible** > **Data Exchange**.
3. Click **...** in the **Direction** column for the tag and set up the desired data exchange direction.
4. Go to the **Triggers** tab and add a trigger tag with a time interval for data exchange. You may also select a tag that triggers exchange when its value is above 0. For more information on triggers, see Triggers.
5. Return to the **Tags** tab and select trigger in the **When** column.

16.11.2. Set Up Data Exchange with Multiple Controllers

Instructions

1. In the Tags Configuration Page, go to the **Controllers** tab.
2. Enter short names for the controllers in the **ID** column, and ensure that all controllers are **Active**.
3. Go to the **Tags** tab.
4. Enable **Columns Visible** > **Data Exchange**.
5. For each tag requiring data exchange, enter the addresses in the corresponding column for all relevant controllers.

IMPORTANT
Ensure that the address is valid for each controller.

6. Click **...** in the **Direction** column for the tag and set up the desired data exchange direction.
7. Go to the **Triggers** tab and add a trigger tag with a time interval for data exchange. You may also select a tag that triggers exchange when its value is above 0. For more information on triggers, see Triggers.
8. Return to the **Tags** tab and select trigger in the **When** column.

NOTE
If a **poll group** includes many tags that change frequently, the time required to process these changes may exceed the set poll time, potentially causing issues with data exchange. To avoid this, either increase the Poll Group Interval or use a time trigger to manage data exchange more effectively. See Poll Groups.

16.12. Aliases

An alias acts as a property of a screen and serves as a stand-in for a tag. It can be used to reference a tag wherever that tag is applicable. However, aliases cannot be used in foreground and background screens, as only tags are allowed in these cases.

When an alias is used, iX Developer treats it as if the original tag were used. Each screen that uses aliases can be configured with a different instance and different tag references.

Using aliases helps in reusing screens without needing to create duplicates.

16.12.1. Create Aliases

Instructions

1. Click **Aliases** in the the lower left part of the Editor Area to open the Aliases View Mode.
2. Click **Add** to include a new alias in the list.
3. Configure the alias settings.

Alias settings

Parameter	Description
Alias Name	The symbolic name of the alias, using any alphanumeric string that begins with a letter (a-z, A-Z).
Default Value	Sets the tag to associate with the alias. If no tag is selected, the alias acts as a local screen variable. The **Default Value** will serve as the alias's default reference in created instances. For more information, see Instances.
Data Type	Sets the data type. This will match the data type of the selected **Default Value** by default. If this column isn't visible, select **Columns Visible** > **Data Type**.

16.12.2. Instances

The **Instances** tab displays all aliases created in the project. You can modify alias values here to override the default values set in the **Aliases** tab, see Create Aliases.

When you change an alias value in the **Instances** tab, the new value appears in **bold**, indicating it differs from the default. To revert to the default, delete the modified value. To change the default value itself, go to the **Aliases** tab.

16.12.2.1. Use the Show Screen Action with Instances

The **Show Screen** action (see Available Actions) allows you to display a specific screen. For screens with aliases, you can also select which instance of the screen to use. This selection determines which set of tags will be assigned to the screen's aliases.

The **Default** option represents a virtual instance configured with the default values of the aliases.

16.12.3. Export and Import Aliases and Instances

You can export or import aliases and instances using the **Import** button in the **Instances** tab of the Tags Configuration Page. This process follows a similar procedure as described in Import and Export Tags.

Export

- When you export alias instances, the default values for each alias appear in a row labeled **DefaultValue** in the exported file.

Import

- If the **DefaultValue** row is present in the import file, you can use it to update alias default values.
- To add new alias instances, add a row in the import file. To create new aliases, add new columns.
- When importing and updating default values for aliases, use the following options to handle conflicts:
  - **Overwrite** or **Merge**: Use these options to update existing alias default values and manage any conflicts in names.
  - **Change**: If selected for **DefaultValue**, the default value will not be updated. Instead, a new alias instance with the modified name will be created.

16.12.4. Aliases Limitations

- Aliases do not support Array Tags.
- Aliases can not be used with Expressions.
- Aliases can not be used with the Trend Viewer Object.
- Aliases can not be used on a background screen, see Screen Group.
- There is no scripting support for a value change event on an alias.

16.13. Tag Format and Data Type

The following tag formats (data types) are available for a tag, provided that the selected driver or OPC server supports the tag format:

Format	Data type	Range
-	DEFAULT	Treated as INT16
Bit	BIT	0 and 1
Bit	BOOL	Values: “false” (0) or “true” (1)*
Signed 16-bit	INT16	-32,768 to +32,767
Unsigned 16-bit	UINT16	0 to +65,535
Signed 32-bit	INT32	-2,147,483,648 to +2,147,483,647**
Unsigned 32-bit	UINT32	0 to +4,294,967,295**
Float with exponent, 64-bit	DOUBLE	1.7 × 10308*
Float with exponent, 32-bit	FLOAT	±3.4 × 1038*
7 × Signed 16-bit	DATETIME	An analog numeric object can present the time format.* Time format: **<yyyy-MM-dd hh:mm:ss>** **DATETIME syntax** • `Dx`, seconds, 0–59 • `Dx+1`, minutes, 0–59 • `Dx+2`, hours, 0–23 • `Dx+3`, day, 1–31 • `Dx+4`, month, 1–12 • `Dx+5`, year, 100–9999 • `Dx+6`, weekday, 1–7
String	STRING	Storing character strings in tags; size can be selected.

* **Controller Representation**: The representation of these types may depend on the driver software.

** **32-bit Value Resolution:** Both iX Developer and the HMI panel can handle up to six decimal digits for 32-bit integer values. Remaining digits may be truncated or set to zero.

16.13.1. C# Type Representation

For use in Scripts, the following table shows the type representation in **C#**:

C# type	Data type	Description
-	DEFAULT	Treated as INT16
`uint`	BIT	C# does not have a “bit” type. In the controller, this is a bit.
`bool`	BOOL	C# `bool` type with values “false” (0) and “true” (1)
`int`	INT16	16-bit integer, -32,768 – +32,767
`uint`	UINT16	16-bit unsigned integer, 0–+65,535
`int`	INT32	32-bit integer, -2,147,483,648 – +2,147,483,647
`uint`	UINT32	32-bit unsigned integer, 0 – +4,294,967,295
`float`	FLOAT	32-bit floating-point number, ±3.4 × 1038

C# type	Data type	Description
`double`	DOUBLE	64-bit floating-point number, ±1.7 × 10308
`DateTime`	DATETIME	Uses 7 elements to store time
`string`	STRING	Storing character strings in tags

16.14. Tag Data Quality

Tag data quality are categorized as **Good**, **Unknown**, or **Bad**.

For internal tags

- Tag data quality will always be **Good**.

For OPC UA

- **Unknown** indicates an issue with a secondary connection line.
- **Bad** indicates a direct connection issue.

When a tag's data quality is **Unknown** or **Bad**, the latest value with **Good** quality will be displayed until the connection is restored. This behavior follows the Unified Automation specifications.

Example setup

	iX App	HMI panel
Connection	[OPC UA Client] ← A → [OPC UA Server]	[Modbus Master] ← B → [Modbus Slave]

Communication errors

- If there is a communication error (**A**) between the OPC UA Server and Client, the tag quality will be **Bad** in the iX App.
- If there is a communication error (**B**) between the Modbus Master driver and Modbus Slave, the HMI panel will show a communication error notification. The tag quality will be **Unknown** in both the HMI panel and the iX App.

17. Scripts

In Script View Mode scripted actions can be configured for screens, objects, tags, function keys etc. Scripts are written in **C#** syntax and provide a flexible way to implement custom functionality.

NOTE
In **iX Developer 3**, the codebase is updated to **.NET 8**. Ensure your scripts are compatible with **.NET 8**.

CAUTION
Incorrect code entered in Script View Mode or Xaml View Mode may result in errors that are not possible to recognize during analysis and build. Erroneous code may result in unpredictable behavior and loss of data.

17.1. Scripting Guidelines and Best Practices

TIP
For a list of keyboard shortcuts, see Keyboard Shortcuts.

Before you begin

- Close any active **Visual Studio** debug sessions.

Scripting Rules

- **Action precedence** – Actions override script logic.
- **Avoid multiple triggers** – Using several triggers can cause errors.
- **Choose one method** – Avoid combining scripts and Action Triggers.
- **Blocking scripts** – Do not use scripts that pause for user input or system resources.

Script events

- Expand object nodes in Script View Mode to display available events.
- Double-click an event to insert its method header.
- **Event limitations**
  - `ValueOff` / `ValueOn` – Not triggered for internal variables during runtime.
  - `ValueChangeOrError` – Only works with controller tags.

External script editing

- Cross-references are not updated during project build.
- Remove Unused Tags may wrongly mark tags used in external scripts.
- To update changes, open the script in iX Developer, make a minor edit, and save.

Memory management

- Avoid using static state to prevent memory leaks.
- Always unsubscribe from events when no longer needed.
- Stop timers properly to avoid execution issues.
- Do not rename screens or objects used in scripts while in Layout View Mode, as it may break the script.

File write safety

- If power is lost during a file write, the file may become corrupted. File writes use caching for performance, which increases the risk. To minimize this risk when scripting, write directly to disk. One way to do this is by using the `WriteThrough` option, see example below.

Example: Write to disk using WriteThrough

```
void WriteFile(string filePath, string fileContent)
{
    const FileOptions option = FileOptions.WriteThrough;
    using (FileStream stream = new FileStream(filePath, FileMode.Create, ▶
FileAccess.Write, FileShare.Write, 4096, option))
    {
        stream.Write(Encoding.UTF8.GetBytes(fileContent));
    }
}
```

17.2. IntelliSense and Script Assistance

Activate IntelliSense

- Press **Ctrl** + **Spacebar** to trigger manually.
- Type a period (.) after a code element for automatic trigger.

Scope access

- Use **Globals** to access items outside the current scope.
- Scripting across different screens is not supported.

Method completion

- Methods are completed without argument lists.
- Manually add parentheses and arguments.

Script errors

- Invalid code may not be detected during build.
- Errors can cause unexpected behavior or data loss.
- Double-click entries in the Error List Window to go to the faulty code.

IntelliSense Limitations

- Not all Tags are shown.
- Tags or objects with names similar to script commands may be hidden.

Tool tips for overloaded methods

- Press **Ctrl** + **Shift** + **Spacebar** to display tool tips.
- Use **Up Arrow** or **Down Arrow** to scroll through options.
- Press **Esc** to close the tool tip.

Type casting

- Use explicit casting for tag formats in overloaded methods.

17.3. Debug Script

Use the **Debug** command in the Project Ribbon Tab to check your C# script code and start an external debugger. The command validates the script file and runs the debugger using the generated script. You can set breakpoints and use other debugging tools. To adjust debugger settings, go to **Script Debugger options** in iX Developer Options.

TIP

If **project validation fails**, check that the absolute path to the project, including its components, does not exceed 260 characters.

17.4. .NET SDK Version Match

iX Developer comes with a fixed version of the .NET SDK.

When you Debug Script with an external debugger, such as Visual Studio, the SDK version depends on what is installed in Windows. Different SDK versions may cause debugger errors.

To use the same SDK version in both environments:

1. Find the SDK version used by iX Developer:
   `%ProgramFiles%\Beijer Electronics AB\{iX Developer Version}\Bin\dotnet\x64\sdk`
2. Install the same SDK version in Windows.

17.5. Script-Based Object Formatting

- Select an object attribute to see tool tips with supported methods.
- Methods outside the default C# API must be fully qualified or included as libraries.

17.6. Script File Paths

Path	Description
`Globals.Environment.Application.CommonApplicationData`	Use this path in scripts to access application data. Points to: `ProgramData\OS3 Packages\{iX_folder}`. Files created by scripts may only be stored here on the internal flash.
`Globals.Environment.Application.StorageLocationIsSdCard`	Returns `true` if the application stores its data on an SD card.

Path	Description
`Globals.Environment.Application.FirstSdCardDriveLetter`	Returns the drive letter of the SD card.
`Globals.Environment.Application.FirstSdCardApplicationPath`	Returns the full path to the application folder on the SD card.
`Globals.Environment.Application.FirstDetectedUsbDriveLetter`	Returns the drive letter of the first detected USB drive.

17.7. Disable a Script

Instructions

1. Right-click the node containing the script.
2. Select **Unhook Event Handler** to disable the script without deleting it.

17.8. Tags in Scripts

To ensure that tags used in scripts are properly detected, include the `IScriptTag` interface in the argument list for the `ScriptModule`. For direct tag references in scripts or expressions, use the syntax: `Globals.Tags.TagName`

External editor considerations

- When using an external editor (e.g., Visual Studio), cross-references will not update when building the project in iX Developer. As a result, the Remove Unused Tags feature may incorrectly mark script-used tags as unused.
- To make sure iX Developer recognizes changes from external editors:
  1. Open the script file in iX Developer.
  2. Make a minor edit.
  3. Save the project.
  4. Build the project.

Example script

```
// ------------------------------------------------------------
// Press F1 to get help about using script.
// To access an object that is not located in the current class, start the ▶
call with Globals.
// When using events and timers be cautious not to generate memory leaks,
// please see the help for more information.
// ------------------------------------------------------------

namespace Neo.ApplicationFramework.Generated
{
    using System.Windows.Forms;
    using System;
    using System.Drawing;
    using Neo.ApplicationFramework.Tools;
    using Neo.ApplicationFramework.Common.Graphics.Logic;
```

```
    using Neo.ApplicationFramework.Controls;
    using Neo.ApplicationFramework.Interfaces;

    public partial class ScriptModule1
    {
        /// <summary>
        /// This ScriptModule method,
        /// sums the int-value of the provided tag arguments.
        /// The tags involved must be activated.
        /// </summary>
        /// <param name="arg1"></param>
        /// <param name="arg2"></param>
        /// <param name="result"></param>
        public void Sum(IScriptTag arg1, IScriptTag arg2, IScriptTag result)
        {
            result.Value = arg1.Value.Int + arg2.Value.Int;
        }
    }
}
```

17.9. NuGet Packages in Scripts

Third-party components and objects added through **NuGet Packages** are available for scripting. Their methods and properties are included in the name completion list, similar to built-in components.

TIP

For more information about NuGet, refer to the NuGet Packages section.

17.10. Scripting Example – Create a Gradient Fill

Task: Fill a rectangle object with a color gradient from red to purple.

Example code implementation

The following code creates a gradient from red to purple in the rectangle. The library containing the `BrushCF()` method is added using the `using` directive at the beginning of the example, along with default libraries.

```
public partial class Screen1
{
    void Screen1_Opened(System.Object sender, System.EventArgs e)
    {
        Rectangle1.Fill = new BrushCF(System.Drawing.Color.Red,
                                      System.Drawing.Color.Purple,
                                      FillDirection.Center);
    }
}
```

18. NuGet Packages

NuGet is a package manager for .NET that simplifies the process of adding, managing, and updating libraries (known as packages) in your projects. It acts as a centralized repository where you can access prebuilt code libraries, tools, and frameworks, enabling you to quickly integrate functionality into your application without having to build it from scratch.

In **iX Developer 3**, NuGet Packages replaces the old **Reference Assemblies** setup for managing project dependencies. This change brings many advantages, including:

Streamlined dependency management	NuGet makes it easy to add, update, or remove libraries in your project. This eliminates the need for manually adding files or handling compatibility issues.
Access to a vast ecosystem of libraries	NuGet provides access to thousands of libraries created by Microsoft, open-source communities, and developers worldwide. These libraries can save time and effort by offering prebuilt solutions for common tasks like database access, communication protocols, or advanced UI components.
Safer version control	Easily update or revert library versions to maintain compatibility and project stability.
Simplified updates	You can quickly update to newer library versions through NuGet, ensuring your project benefits from security patches, bug fixes, and performance improvements.
Better organization	Dependencies are automatically tracked in a project file, making it easier to maintain and share your project with others.

NOTE
Beijer Electronics does **not** provide support for scripting.

IMPORTANT
The iX App is a commercially licensed software. Users must ensure that any third-party libraries or components used in combination with the iX App comply with its licensing terms.

CAUTION
Using NuGet packages may expose your project to various risks, including legal, security, and application quality concerns. It is the user's responsibility to manage these risks appropriately.

18.1. Manage NuGet Package Security

When using **NuGet** in your projects, it's essential to consider potential security risks. Follow these guidelines to ensure a secure and reliable implementation:

Verify the source	NuGet packages vary in quality and origin. Some may come from untrustworthy publishers or include vulnerabilities. Always

	check the publisher's identity. Prioritize packages from recognized sources, such as **Microsoft**, reputable organizations, or well-regarded community developers.
Review package popularity and updates	Packages with higher download counts and regular updates are usually better maintained and more reliable. Use the **NuGet Package Manager** to examine a package's download statistics and version history.
Check version notes and change logs	Review recent version notes and change logs for insights into fixes and updates.
Avoid outdated packages	Older packages may lack support for the latest versions of **.NET** or have unresolved security issues. Regularly update your project's NuGet packages to align with current standards and security requirements.
Ensure compatibility	For details, see NuGet Package Compatibility
Audit dependencies	Using excessive external libraries increases the risk of vulnerabilities and maintenance issues. Limit your use of external packages to those that are absolutely necessary. Perform regular dependency audits to confirm security and relevance.

By following these guidelines, you can minimize risks and ensure a secure, well-maintained project.

18.1.1. NuGet Package Compatibility

iX Developer includes functionality to detect and warn about compatibility issues between NuGet packages or between packages and iX Developer itself. These checks occur during project build, and any issues are listed in the Error List Window.

Best practices

Avoid conflicting packages	Do not install packages that conflict with those used by the iX App.
Use consistent versions	Always install the same version if a package is already in use.
Respect .NET Compatibility	Do not install packages that require **.NET 9** or later. Packages supporting only the old **.NET Framework** are also not supported.
Clear NuGet cache	Close iX Developer and clear the local NuGet cache to resolve issues caused by locked or incomplete files.
Visual Studio Users	Click **Clear All NuGet Storage** in **Visual Studio** to clear the NuGet cache.

Common compatibility issues

NModbus errors	Installing **NModbus 3.0.74** alongside **NModbus.Serial 3.0.78** results in a conflict because **NModbus.Serial** requires **NModbus 3.0.78**. NU1605: Detected package downgrade: NModbus from 3.0.78 to 3.0.74. Reference the package directly from the project to select a different version. NU1605: Project68 -> NModbus.Serial 3.0.78 -> NModbus (>= 3.0.78) NU1605: Project68 -> NModbus (>= 3.0.74) **Solution**: upgrade **NModbus** to **3.0.78** to align versions.
MimeKit error	Installing **MimeKit 4.0** can cause problems if iX Developer uses **MimeKit 3.6** internally. Project output assembly [MimeKit, Version=4.0.0.0, Culture=neutral, PublicKeyToken=bede1c8a46c66814] does not match the default runtime project assembly [MimeKit, Version=3.6.0.0, Culture=neutral, PublicKeyToken=bede1c8a46c66814] Caused by NuGet package dependency chain: MimeKit 4.0.0 **Solution**: Match the version used by iX Developer.

18.1.2. License Compliance for NuGet Packages

When using NuGet packages in **iX Developer**, follow these guidelines to ensure proper license management:

Guidelines

Ensure license compliance	Review and comply with all license terms associated with the NuGet packages you install.
Verify license compatibility	Confirm that the licenses of all integrated packages are compatible with one another to avoid potential conflicts.

IMPORTANT
The end user is responsible for reviewing and ensuring adherence to all license requirements.

18.2. Supported NuGet Reference Formats

iX Developer supports commonly used NuGet package reference formats, provided they include the full package name and version.

Examples of supported formats

- Serilog 2.12
- dotnet add package Serilog --version 2.12.0
- <PackageReference Include="Serilog" Version="2.12.0" />
- NuGet\Install-Package Serilog -Version 2.12.0
- microsoft.net.sdk.maccatalyst.manifest-6.0.200.15.2.302-preview.14.122

18.3. Install NuGet Packages

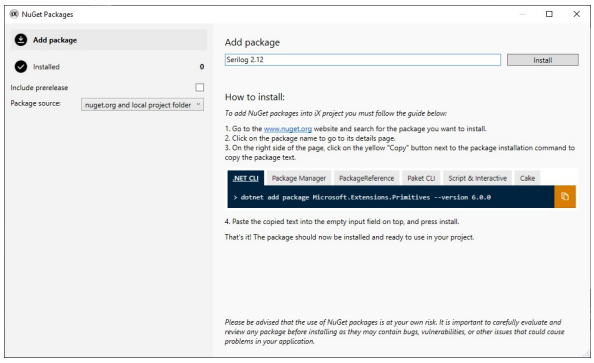
IMPORTANT
Before installing a NuGet package, ensure it is secure and trustworthy. For details, see Manage NuGet Package Security

VIDEO TUTORIAL - INSTALL NUGET PACKAGES

https://player.vimeo.com/video/1057628015?h=971fbda90e&badge=0&autopause=0&player_id=0&app_id=58479

Instructions

1. Click **Project** > **NuGet Packages** to open the **NuGet Packages Manager**,.
2. Follow the **How to Install** instructions in the **NuGet Packages Manager**.



18.3.1. NuGet Package Source Options

In the **NuGet Packages** window, use the **Package source** dropdown to select a source:

Local project folder

Manually place `.nupkg` files in the `NuGetSource` folder in the Project Folder. This makes the package available for installation. To install the NuGet package, see Install NuGet Packages.

> **NOTE**
> The `.nupkg` file must remain in the `NuGetSource` folder while the package is in use.

nuget.org and local project folder

Combines the local folder with nuget.org, the main online NuGet repository with over 300,000 packages.

All

Searches all available sources, including:

- Local project folder
- nuget.org
- Any additional sources defined in the system or user NuGet settings

> **TIP**
> Additional sources can be managed using the **Visual Studio Package Manager**. For more details, see Manage packages with the Visual Studio Package Manager Console (PowerShell).

18.3.2. Install NuGet Packages Offline

If the development PC is offline, you can install NuGet packages using the local `NuGetSource` directory in the project.

Instructions

1. Before installing a NuGet package, ensure it is secure and trustworthy. For details, see Manage NuGet Package Security
2. Copy the NuGet packages to the local `NuGetSource` directory.

3. Follow the steps in Install NuGet Packages and set the Package Source to the **Local Project Folder** to avoid delays from external repository lookups.

BUILD ERRORS DUE TO MISSING .NET RUNTIME PACKAGES

iX Developer may occasionally require an Internet connection. If build errors appear due to missing .NET runtime packages, withe the Package Source set to the **Local project folder** , temporarily switch to **nuget.org and local project folder** and connect to the Internet.

Common .NET runtime packages required

- `Microsoft.NETCore.App.Runtime.win-x64`
- `Microsoft.NETCore.App.Runtime.win-arm64`
- `Microsoft.WindowsDesktop.App.Runtime.win-x64`
- `Microsoft.WindowsDesktop.App.Runtime.win-arm64`
- `Microsoft.AspNetCore.App.Runtime.win-x64`
- `Microsoft.AspNetCore.App.Runtime.win-arm64`

Alternatively, you can manually download the packages and place them in the `NuGetSource` directory of the project.

18.4. View Installed NuGet Packages

Instructions

1. Click **Project** > **Project** > **NuGet Packages** to open the **NuGet Packages** window.
2. Click the **Installed** tab to view a list of installed packages, including:
   - Package name
   - Version
   - Description

TIP

If a newer version is available, it will be shown in green below the installed version.

3. To display prerelease versions, select **Include prerelease**.

NOTE

Prerelease versions may be less stable than official releases.

4. Use the NuGet Package Source Options to filter the packages.

18.5. Uninstall, Upgrade or Downgrade NuGet Packages

To manage NuGet packages in your project:

1. Click **Project** > **Project** > **NuGet Packages** to open the **NuGet Packages** window.
2. To **uninstall**, click **Uninstall**,
3. To **upgrade** or **downgrade**, select the desired version from the **Version** dropdown list and click **Install**.

NOTE
An internet connection is required to access versions not stored locally. If working offline, ensure the required versions are available in the **Local Project Folder**.

CAUTION
Outdated packages can be a security risk. Keep your libraries up to date!

18.6. IntelliSense Support for NuGet Packages

The script editor supports **IntelliSense**, which provides code completion and information for types contained in NuGet packages. After adding new packages and building the project, IntelliSense becomes available for those package types.

Limitations

- Supported for packages targeting **.NET Framework 4.8 (NET48)** or **.NET Standard 2.0** and earlier.
- Not supported for packages containing only **.NET 8** or later DLLs.

18.7. Training: Create, Install and Use a NuGet Package

This chapter demonstrates how to create a NuGet package to facilitate interaction between a Button Object and a Text Object.

Process

1. Create the NuGet Package: Develop a DLL containing the desired method, then package the DLL as a NuGet package.
2. Install the NuGet Package: Add the NuGet package to your iX Developer project.
3. Use the NuGet Package: Call the method from the DLL and display the result in a text box within your project.

18.7.1. Create the NuGet Package

Instructions

1. Open a command line prompt and run the following to create a new library project:

   ```
   dotnet new classlib --name Library1
   ```

   This creates a folder named `Library1` containing `Library1.csproj` and `Class1.cs`.

2. *(Optional)* **Add .NET48 support**: In Visual Studio Open `Library1.csproj` and edit the project file. Change `<TargetFramework>net8.0</TargetFramework>` to:

```
<TargetFrameworks>net8.0;net48</TargetFrameworks>
```

Remove any properties incompatible with **.NET48**.

3. Create a public method in `Class1.cs` that returns an increasing value with each call.

```
namespace Library1
{
    public class Class1
    {
        // Before change: (empty class body)

        // After change:
        private int _number = 0;

        public int ReadNumber()
        {
            return _number++;
        }
    }
}
```

4. Ensure that the project compiles successfully in Visual Studio.
5. Navigate to the project directory in the command line:

```
cd Library1
```

6. Run the following command to create the NuGet package:

```
dotnet pack -c Release
```

7. Upon success, the console will confirm the creation of the **.nupkg** file:

```
Successfully created package ▶
'D:\Projects\Library1\bin\Release\Library1.0.0.0.nupkg'.
```

18.7.2. Install the NuGet Package

1. Open your iX Developer project.
2. Create a Button Object and a Text Object and save the project.
3. In the file system of your new project, find the `NuGetSource` folder.
4. Copy the `Library1.0.0.0.nupkg` file created in chapter Create the NuGet Package to the `NuGetSource` folder.
5. In iX Developer open **Project** > **NuGet Packages**.
6. Set the **Package Source** dropdown to **nuget.org and local project folder**, see NuGet Package Source Options.
7. In the **Add package** text box, enter the name of the package.
8. Click **Install**.
9. **Build** the project to finalize the installation.

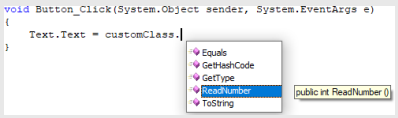
18.7.3. Use the NuGet Package

1. In iX Developer, create a click event for the button and add the following code (in bold).

```
namespace Neo.ApplicationFramework.Generated
{
    using System.Windows.Forms;
    using System;
    using System.Drawing;
    using Neo.ApplicationFramework.Tools;
    using Neo.ApplicationFramework.Common.Graphics.Logic;
    using Neo.ApplicationFramework.Controls;
    using Neo.ApplicationFramework.Interfaces;
    using Library1;

    public partial class Screen1
    {
        private Class1 customClass = new Class1();

        void Button_Click(System.Object sender, System.EventArgs e)
        {
            Text.Text = customClass.ReadNumber().ToString();
        }
    }
}
```

TIP

When using the new class from the NuGet package, notice that IntelliSense provides suggestions for the new method.



2. Run the **iX Developer** project.
3. Click the button object multiple times. The value in the text box will increment with each click.

18.8. NuGet Resources

Here are some valuable resources for creating and consuming NuGet packages:

- Quickstart: Create and publish a NuGet package using Visual Studio (Windows only)
- Quickstart: Create and publish a package with the dotnet CLI
- Create a package using the nuget.exe CLI

19. Language and Translation

iX Developer supports multilingual applications. Text and system texts can be translated directly within the application or exported to a text file for external translation. Once translated, the file can be imported back into the application.

The application language in the iX App can be changed, for example, based on a tag value.

Pre-translated system texts are included with iX Developer. When importing system texts into panel projects, any warnings can be safely ignored.

19.1. Set Up Multiple Languages

The Multiple Languages Function is included by default in iX Developer. Refer to the following chapters for instructions on setting up multiple languages.

19.1.1. Add Languages

Instructions

1. Open the Multiple Languages Configuration Page.
2. Select the **Languages** tab.
3. Click **Add** to add a new language. Each language added is assigned an index number, with index 0 reserved for the default language (the language used when the project was originally created).

TIP
Use the **Show Selection** button to filter the languages visible in the current view, see Filter Items on the Configuration Pages.

4. Select the desired language from the dropdown list in the **Language** column.
5. In the **Use in Runtime** column, specify whether the language should be available in the iX App.
6. In the **Keyboard Layout** column, choose the appropriate layout for the Virtual Keyboard. Note that virtual keyboard sizes may vary depending on the language.

If Scripts are used for texts that are also defined in the **Multiple Languages** function, the text values set by the script may be overridden by the values in the multi language server.

To ensure the script has priority:

- Leave the corresponding entry in the **Text** column of the **Texts** tab blank.
- Include all multilingual alternatives directly in the script.

This setup prevents conflicts between scripted values and the values set in the **Multiple Languages** configuration.

19.1.2. Configure the Multiple Language Font Template

Font templates allow you to assign different fonts and font scaling for specific languages, overriding most project texts in the iX App. When the language is switched, the texts on screen will adopt the font and scaling associated with that language. To activate the **Multi Language Font Template**, follow these steps:

1. Open the Multiple Languages Configuration Page.
2. Go to the **Languages** tab and click **Settings** to open the **Properties** window.
3. Enable the checkbox **General** > **Font Template Settings** > **Enable Font Templates**.
4. When **Enable Font Templates** is enabled, two new columns will appear in the **Languages** tab:
   - **Font**: Choose a font from the list of installed fonts on the development PC.
   - **Font Scaling**: Adjust the font size from -99% to 500%, relative to the iX Developer values. Positive values increase the font size, while negative values reduce it.

Limitations

- If objects in the project use **dynamic font sizing**, the dynamic settings will take precedence, and font templates will not affect these objects in the iX App.
- Switching the design language will cause the **Font** and **Font Scaling** settings in the **Languages** tab to have no effect.

IMPORTANT
The user is responsible for ensuring that the selected font and scaling are suitable for the project.

19.2. Translation

Text and system texts can be translated directly within the application or exported to a text file for external translation. Once translated, the file can be imported back into the application.

19.2.1. Export Languages

Languages in iX Developer can be exported to a text or Excel file for translation.

Instructions

1. Go to the Multiple Languages Configuration Page.
2. Navigate to the **Texts** tab, **System Texts** tab or **Text IDs** tab.
3. Click the small arrow next to the **Import** button and select **Export**.
4. Specify the export parameters:

Parameter	Description
Format	Choose format: **Text file** or **Excel file**.
Path	Select the location where the file will be saved.
Delimiter	For text file exports, choose a delimiter from the options: **comma**, **semicolon**, **space**, or **tab**.
Languages	Select the languages to include in the export. The default language is always included.

5. Click **OK**.

19.2.2. Import Languages

After translating or editing the exported language file, you can re-import it into iX Developer.

IMPORTANT
Ensure you create a backup of the project before importing languages to avoid potential mismatches.

Instructions

1. Go to the Multiple Languages Configuration Page.
2. Navigate to the **Texts** tab, **System Texts** tab or **Text IDs** tab.
3. Click the **Import** button.
4. Specify the import parameters:

Parameter	Description
Format	Choose format: **Text file** or **Excel file**.
Path	Browse to the location of the file to import.
Languages	Select the languages to import. The default language is always included.
Import Strategy	**Key**: Imports based on the **Designer** and **Object** columns in the **Texts** tab of the Multiple Languages Configuration Page. Recommended if the default language text has changed after export. **Reference Text**: Imports based on the **Text** column in the **Texts** tab of the Multiple Languages Configuration Page. Recommended if the default language text has changed after export.

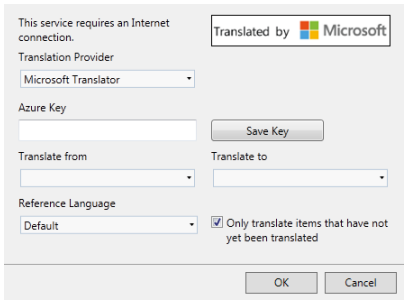
5. Click **OK**.

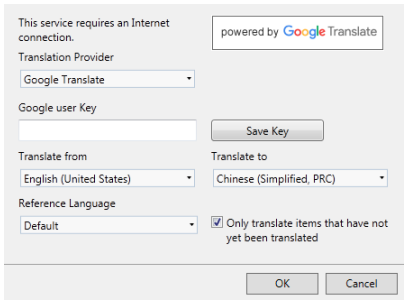
19.2.3. Automatic Translation Tool

iX Developer includes an automatic translation tool that uses Microsoft Translator or Google Translate to quickly translate user-defined or system texts. An account with at least one of these providers is required to obtain the necessary API keys. Each language is translated individually.

IMPORTANT
Automatic translations provide limited functionality, especially for phrases or synonyms.

To open the translation tool, open the Multiple Languages Configuration Page, go to the **Texts** tab or **System Texts** tab and click **Translate**.





Settings

Parameter	Description
Translation Provider	Select the desired translation provider.
Azure Key *(only in Microsoft Translator)*	Enter the Azure key from your Microsoft Azure Translator API subscription.
Google user Key *(only in Google Translator)*	Enter the public API key from Google Developers.
Save Key	Save the key to your user profile on the development PC.
Translate from	Select the source language.
Translate to	Select the target language.
Reference Language	Set the language in which the project was originally created.
Only translate items that have not yet been translated	Enable to skip items already translated.

NOTE
Translation speed may vary depending on server load and network conditions.

19.3. Edit System Texts

You can edit system texts in the **System Texts** tab of the Multiple Languages Configuration Page.

Pre-installed translations are available for the following languages:

- English
- German
- French
- Brazilian Portuguese
- Chinese (Traditional)
- Chinese (Simplified)

By default, regional expressions for time, date, and weekdays match the settings of the system account. However, these can be modified to some extent in the Time Zone and Region Group.

19.4. Edit User Texts

You can edit user-defined texts in the in the **Texts** tab of the Multiple Languages Configuration Page. Texts in the default language can also be edited.

If an object is linked to a text in the Text Library Function, , the object's internal name (not shown in the iX App) will appear in grey in the **Texts** list. This indicates that changes will not affect the iX App.

19.5. Text IDs

TextIDs are unique identifiers used for managing text strings within a project—especially useful for multi-language support.

What is a Text ID?

- A TextID is a reference ID tied to a specific string of text.
- It allows you to display dynamic or static text in various controls (labels, buttons, alarms, messages) without hardcoding the actual text.
- You assign a TextID to a control, and it will show the associated string, based on the current language setting. Once a text is linked to a Text ID, the object will display the corresponding text only. The Text ID becomes hidden and cannot be edited.

Why use Text IDs?

- **Localization**: Easily switch languages—change the language and all texts update automatically.
- **Centralized management**: All text resources are stored in one place (the Text Library Function), making it easier to edit or translate.
- **Consistency**: Reuse the same TextID across multiple screens to maintain consistency.

19.5.1. Add Text ID to an Object

Instructions

1. Open the Multiple Languages Configuration Page.
2. Go to the **Text IDs** tab and enable **Text IDs enabled**.
3. Select the object.
4. Go to **General** > **Text**. Click the down-arrow next to the **text field** to open the **Text ID browser**.

> **TIP**
> The text field now filters the Text ID list.
> **Format**: `Text ID – Text` (where `Text` is the default language text).

5. *(Optional)* To configure a new Text ID:
   a. Click **Add**.
   b. Enter the default language text and the translated text(s).

> **TIP**
> Only the default language is shown initially. Right-click the column header to display additional languages.

   c. Click **OK**.
6. Select the desired Text ID and click **OK**.

The object's displayed text now updates when changing the language in iX Developer or the iX App.

Example

Text ID	Default Language	Spanish
1	OK	Aceptar

Assigning Text ID **1** to a button shows **OK** in English and **Aceptar** in Spanish.

19.5.2. Configure Dynamic Text IDs

A **dynamic text ID** is a unique identifier linked to a dynamic text string. When configured, the tag value in the iX App represents a **Text ID**. The displayed text corresponds to the current language's translation for that key. If the tag value is not found in the **Text ID** table, an empty string is displayed.

Instructions

1. Add a Text ID to an object as described in Add Text ID to an Object.
2. Click the **Dynamic Text ID** tab. Select wether to link the text to a
   - **Tag**, or
   - **Alias**.
3. Configure the following settings:

Parameter	Description
Default Value Used in Design Mode	Displays the value in iX Developer but not in the iX App.
Add	Adds a tag or alias.
Reset	Removes the **Dynamic Text ID** connection and restores the original text.

20. Security

This section describes the security functions of iX Developer and the iX App.

CAUTION

Security measures—such as user access control, password policies, network protection, firewalls, and antivirus software—should be evaluated and implemented based on a risk assessment tailored to your specific installation and application.

The user is responsible for:

- Conducting the security risk assessment.
- Implementing, configuring, and maintaining security protocols.
- Ensuring proper deployment and management of iX Developer in accordance with organizational policies to maintain functionality and compliance.

20.1. Security Guidelines

This section provides guidelines for securing both the network and physical aspects of your iX Developer environment. Proper firewall configuration, antivirus protection, and physical security measures are essential for safeguarding your system from unauthorized access and ensuring reliable and secure operation. By following these practices, you can significantly reduce potential security risks associated with both network and hardware vulnerabilities.

20.1.1. Project Back Up and App Transfer Security

For project backup and app transfer best practices, see Back Up the Project and Transfer, Fetch, Export and Import.

20.1.2. Email Security

If **Authenticated Login** in the Configure Distribution Devices Window is selected, an email from the Alarm Distributor Function requires a username and password for protection.

20.1.3. Password Security

Password protection for the Web Server and OPC UA Server is configured in iX Developer during project creation. Any changes to passwords must be made in iX Developer, and the updated project must be re-downloaded.

Password protection for Users and Email Security can be set both in the iX App and in iX Developer.

NOTE

By default, Functions have no passwords. It is highly recommended to set a password when enabling a new function, especially if it is accessible over the internet.

20.1.3.1. Password Guidelines

- Keep passwords confidential. Do not share them.
- Change your password immediately if you suspect it is compromised.
- Use passwords with at least eight characters; Beijer Electronics recommends fourteen or more.
- Avoid common words or patterns. Choose a strong password that is hard to guess—even by someone who knows you.
- Sign out when you leave the device unattended.
- Do not reuse the same passwords in OS3 and iX Developer.

20.1.4. OPC UA Security

At the first OPC UA Server startup, a Self-Signed Certificate is automatically generated. This certificate is valid for 20 years but does not contain an IP address, as the IP could change after the certificate is generated.

When re-downloading a project to a panel with an existing OPC UA certificate, you can:

- Keep the `pki` folder to retain the certificate, or
- Delete it to generate a new one.

Limitations

- **Discovery**: Not supported.
- **Encoding/Transport**: Only UATCP Binary supported
- **User authentication**: Only username and password, one user account
- **Security**: No encryption support.

20.1.5. Antivirus Program

It is recommended to have an antivirus program installed on your development PC.

20.1.6. Physical Protection

Physical security measures are essential to prevent unauthorized system access. These may include locks, access control, surveillance, and restricted access to equipment.

The iX App runs on both virtual and physical machines. Removable assets and communication lines may be used in both environments.

CAUTION
Users are responsible for implementing appropriate physical protections to secure the system, removable media, and communication lines.

20.1.7. Firewall Settings

- It is recommended to configure network firewalls to limit access to the HMI panel(s).
- Ensure that Open Ports comply with your organization’s internal security standards.
- For detailed setup guidance, contact your IT administrator.

20.2. Self-Signed Certificate

In iX Developer, a self-signed certificate is used to secure communication when transferring a project from a development PC to an HMI panel. This certificate ensures encrypted data transfer, protecting sensitive information during the deployment process.

A self-signed certificate is a digital certificate signed by the entity creating it, rather than a trusted Certificate Authority (CA). While it enables secure encryption, it is not automatically trusted, and might trigger security warnings.

The first time a project is transferred to an HMI panel, ensure the **Accept self-signed certificate** checkbox is selected. For more information, see Transfer the iX App to the HMI Panel.

20.3. Open Ports

This table describes the open ports on an **X3 HMI panel**. Additional ports may open dynamically in OS3 based on your iX Developer project.

Port number	Type	Name	Description	Open by default	Can be closed
443	TCP	HTTPS	HTTPS interface for app transfer and API access.	Yes	No

NOTE

Communication drivers may use either dynamic or static ports, depending on the specific driver. Some drivers operate with fixed ports, while others do not. For detailed information, consult the Documentation for the driver in question.

21. Optimize Performance

The following section highlights factors to consider for improving the performance of the iX App on the HMI panel.

KEY RECOMMENDATIONS

- **Development PC**: Follow the Recommended System Specifications for iX Developer.
- **HMI panel**: Use an HMI panel that matches the requirements of your application.

21.1. Recommended Project Item Limits

To ensure optimal performance, keep the number of items within the recommended limits listed below:

Project item	Recommended limit		
	X3 extreme	X3 marine	X3 pro
Tags	4000		
Active controllers	10		
Data loggers	25		
Database items	700		
Alarm items	500		
Alarm viewer rows	200		
Alarm database max. rows	1000		
Screens	500		
Objects on screen	400		

21.2. Maximize the HMI Panel Hard Drive Lifetime

The HMI panel uses a **Flash SSD** (Solid State Drive) for storage. Flash SSDs have no moving parts, providing high reliability and fast performance. However, the lifespan of the drive is determined by the number of read and write operations, which the manufacturer tracks using proprietary algorithms. As the algorithm is confidential, the exact lifetime cannot be calculated, but only the number of reads and writes are published.

To maximize the lifespan of the Flash SSD, minimize the number of write operations. Follow these best practices to reduce the risk of premature wear:

- **Store the database on an SD Card** – To reduce Flash SSD wear, store the database on an SD card, see Use SD Card to Reduce Flash Wear.
- **Limit logging accuracy** – Avoid logging data with higher accuracy than required, such as floating point values.
- **Use hysteresis for fluctuating values** – Reduce high fluctuations by using the hysteresis feature in the connected controller.
- **Separate fast and slow changing values** – Log fluctuating values in one logger and stable values in another.

- **Minimize logging frequency** – Only log at the necessary refresh rate.
- **Disable logging when not needed** – Turn off logging when the machine is idle or not in use.
- **Use default logger setting** – Enable the **Log Changes Only** option in the Data Logger Properties to log only when a value changes.
- **Keep log size small** – Log only the necessary data with the fewest rows possible.
- **Avoid extensive file writing in scripts** – Limit file writing when controlling storage through scripts.

21.2.1. Use SD Card to Reduce Flash Wear

Beijer Electronics **recommends using an industrial SD card** instead of the HMI panel hard drive for database/data logging.

Advantages of using an SD Card

- **Extends panel lifespan** – An hard drive wears out with frequent writes, while an SD card is replaceable.
- **Larger storage capacity** – SD cards generally have more space than the built-in hard drive.

21.3. Manage Storage Space

If storage usage reaches a critical level, the system disables alarm and data logging to prevent malfunction.

Critical storage warning:

99% of disk used. Alarm logging is now disabled. Data logging is now disabled.

- The system tag **Database Error Active** is triggered.
- The **Database Error Message** tag contains the error details.
- These tags are **not reset automatically**. You must reset them manually after resolving the issue.
- Alarm and data logging resume automatically when storage usage drops below the critical level.

Recommendation

- Use the **Max Size** setting in Project Properties > **Database** to define storage limits. When the limit is reached, a warning appears to prompt action.

21.4. Driver Performance

Choose efficient drivers for better communication speed and minimize the number of active drivers to reduce system load. For more information, see Driver List X3 HMI Panels.

21.5. Screen Performance

Improving screen update time reduces CPU load and flash memory usage. This is especially important in projects with high graphic demands.

Recommendations

- **Efficient screen switching** – Use the **Show Screen** action (see Available Actions) triggered by a function key or object for the fastest screen changes.

- **Simplify screen content** – Fewer objects lead to faster loading. Complex elements, such as alarm viewers, can slow down performance. Keep frequently displayed or background screens as simple as possible.
- **Enable screen caching** – Caching improves screen switching speed. You can cache more screens if RAM usage is low. Large screens use more memory, which may prevent smaller screens from being cached.
  - Set a screen's **Cached** property to **False** if caching is unnecessary.
  - Keep RAM usage below 85%. If this limit is exceeded, cached screens may be removed, which slows down screen switching.

TIP
For more information, refer to the Screen Caching chapter.

21.5.1. How to Design Effective Screens?

Start by identifying the required functions and planning the application from a high-level overview down to detailed design.

A project consists of multiple screens containing objects that exchange data with the controller. Arrange screens in a hierarchy for structured navigation or as a sequence for process control. Choose a layout that accurately represents the process and supports efficient operation.

Test the complete application—or individual sections—on the development PC before transferring it to the iX App.

TIP
For detailed information on screens, refer to the Screens chapter.

21.5.2. Screen Caching

Screen caching involves temporarily storing a screen's content in memory to enable faster loading during future use.

21.5.2.1. Screen Events

Screens trigger events when they load and close. This behavior occurs regardless of whether the screen is cached. When a cached screen is replaced, the following sequence occurs:

1. **Deactivate dynamic bindings**: All dynamic bindings are deactivated.
2. **Hibernate active controls**: Controls with running timers or processes are paused.
3. **Trigger close events**: Both the **Closing** and **Closed** events are triggered sequentially.

21.5.2.2. Scripts in Cached Screens

When using Scripts or Actions in a cached screen, ensure that scripts shut down properly to avoid performance issues. For example, if a timer or thread starts during the screen's **Opened** event, make sure to pause, stop, or dispose of it during the **Closing** or **Closed** event.

Improper script management may result in running processes in the background, causing higher CPU usage or other issues.

21.6. Tag Handling

Follow these guidelines to improve tag performance in your iX App:

- **Reduce the number of tags** – Fewer tags reduce data processing and improve performance.
- **Minimize tag packages** – See Minimize Tag Packages.
- **Keep tags in consecutive order** – Define controller tags in a continuous memory range (e.g., M0.0 to M11.7). Scattered tags (e.g., I0.4, Q30.0, M45.3) increase update time. The number of tags per package depends on the driver. See the driver manual for detailed information.
- **Set a longer update interval** – Tags update every 500 milliseconds by default. If some tags do not require frequent updates, group them and set a longer update interval. Configure Poll Groups carefully—frequent updates may reduce system performance.
- **Avoid keeping tags constantly active** – In scripts, link tags to events such as value changes instead of continuous monitoring. Use the **Passive Value Change** event when possible. It only triggers when the tag is actively used. Also see About Static and Dynamic Tags.
- **Limit sampled tags in the Trend Viewer Object and Data Logger Function** – Fewer sampled tags reduce the load on data logging.
- **Other tags that reduce performance more than others**:
  - **Array tags**: see Available Tag Types.
  - **Tags connected to multiple controllers**: see Configure Multiple Controllers.
  - **Tags used in scripts**: see Tags in Scripts.
  - **Static tags**: see About Static and Dynamic Tags.
  - **System tags**: see Available Tag Types.
  - **Non-volatile tags**: see Available Tag Types.
  - **Tags using data exchange**: see Data Exchange.
  - **Tags with the following settings**:
    - Access right *(other than ReadWrite)*
    - Offset
    - Gain
    - Read/Write expression
    - Index register
    - Log to Audit trail

21.6.1. About Static and Dynamic Tags

Tags used for driver communication can be either **static** or **dynamic**. They update based on the following rules:

21.6.1.1. Static Tags

Static tags update continuously, even when not shown on the screen. These tags therefore affect communication time between the HMI panel and the controller.

Static tag types

- Alarm tags
- Data logger tags
- Multiple languages tags
- Controller tags with value change events

Exceptions - Do Not affect communication time

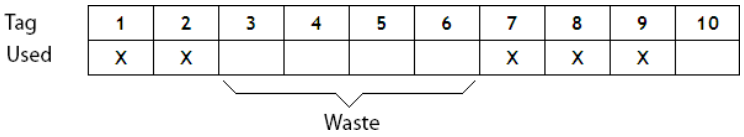
- Alarm messages
- Schedulers
- Tags linked to function keys

21.6.1.2. Dynamic Tags

Dynamic tags update only when displayed on screen. For example, an Analog Numeric Object uses a dynamic tag while it is visible. These tags does not affect the communication time between the HMI panel and the controller as much as **Static Tags**.

21.6.2. Minimize Tag Packages

Tags are grouped into packages before transfer. Fewer packages result in faster communication. Placing tags consecutively reduces the number of packages and improves efficiency.



If consecutive arrangement is not possible, **waste** occurs—unused space between non-adjacent tags in a package. The amount of waste depends on the driver. Refer to your driver's manual for details. See Driver List X3 HMI Panels to locate the correct driver manual.

21.6.2.1. Use Bit-Addressed Words

Using bit-addressed words is more efficient than using regular bit devices. This allows more digital devices in one telegram.

Example

Driver x	Analog signals	Digital signals
Signals/package	29	124
Waste	20	0

A driver supporting 29 analog or 124 digital signals per package can fit 464 digital devices using bit-addressed words (29 × 16), almost four times more than regular bit devices.

21.7. Optimize Object Usage

Follow these best practices to reduce memory use and improve screen update speed when working with objects.

- **Use pictures with optimized sizes** – Smaller images use less memory and load faster.

- **Minimize gradient use to reduce CPU load** – Gradients slightly increase CPU usage. Avoid using many gradient-styled objects, especially when linked to fast-updating tags (e.g., bar graphs).
- **Limit the number of polygons** – Polygons are complex and redraw every time the screen opens. This slows down performance, even with screen caching. Refer to Shapes Objects for more information.
- **Use styled buttons** – Styled buttons perform better than default buttons. Refer to Button Object for more information.
- **Avoid transparency and overlapping objects** – Transparent elements (e.g., circular meters or sliders) reduce performance. Avoid overlaps within their bounding boxes and do not use gradients in backgrounds or dynamic objects on such screens.

21.8. Optimize Alarm Usage

Only use the alarms your project needs. Too many alarms can slow down the system and make the alarm list harder to manage on the HMI panel.

Alarm optimization tips

Tip	Description
Remove unused alarms	Delete alarms that are no longer relevant to reduce memory use.
Group similar alarms	Use categories or priorities to simplify handling and improve readability.
Avoid frequent triggers	Limit alarms that activate often to reduce system load.
Set proper thresholds	Ensure alarms only trigger under meaningful conditions.
Use summary alarms	Replace many minor alarms with one summary alarm where appropriate.
Test alarm logic	Validate alarms during development to avoid overload in runtime.

TIP

For detailed information on alarms, refer to the Alarms section.

21.9. Data Logger Performance

To improve panel performance, reduce the number of active data loggers and adjust logging settings as needed.

General recommendations

- **Group log items** – To optimize memory usage, group all *LogItems* with the same sampling interval into a single data logger. Avoid assigning a separate data logger to each Trend Viewer Object.
- **Adjust logging settings** – Logging affects CPU performance. Set appropriate log frequency, select only necessary tags, and review Data Logger Properties to optimize system load.

Recommended Limits

- Follow the Recommended Project Item Limits to avoid performance issues. Logging more than the recommended number of rows may delay project startup.

21.9.1. Handle High Logging Load

When the logging load exceeds system capacity, the system queues log entries and a warning appears:

Data logging has too high load. Data might be lost.

If the load continues, logging is disabled, and this error appears:

Data logging is overloaded. Data logging is now disabled.

The **Database Error Active** system tag is set, and the **Database Error Message** tag contains the error details.

After fixing the issue, reset both tags. Logging resumes automatically once the queue size returns to normal.

21.10. ASCII String Usage

ASCII strings are transmitted in separate telegrams, which can reduce communication performance if many strings are used.

If an ASCII string only has a small number of different string values, it may be a good idea to use the Text Library Function. This allows predefined string contents, minimizing the impact on driver performance.

22. Troubleshooting

Use this section to resolve common iX Developer problems.

Transition from iX Developer 2 to iX Developer 3

Problem	Possible cause	Solution
Refer to X2 to X3 - Transition Guide for iX Developer.		

Controllers and drivers

Problem	Possible cause	Solution
iX App fails to open and shows “missing drivers” error.	A controller lacks its driver.	Download and install the correct driver. See Update or Install Drivers.
Error states “driver not supported”.	The driver isn’t supported by your iX Developer version.	Open the project in an earlier iX Developer version that supports this driver, then replace it with a supported driver.
		Open the project in an earlier iX Developer version that supports this driver and remove the unsupported controller and driver.
		Wait for future iX Developer updates that support this driver.

Build, Rebuild, Run, Simulate

Problem	Possible cause	Solution
Error "Access to the path..." on Build, Rebuild, Run or Simulate.	Your Project Folder is synced with **Microsoft OneDrive**, **Google Drive**, or similar.	Move your project to a different folder.
		Disable syncing on the project folder.
Build, Rebuild, Run or Simulate fails.	Project path (including components) exceeds 260 characters.	Shorten file and folder names to reduce path length.
Error indicates "an expression cannot be found".	Objects from the Component Library Window use non-global expressions.	Insert objects that use only global expressions. For more information, see Expressions.

Objects

Problem	Possible cause	Solution
Error states "an invalid name is used".	You used a reserved system device name.	Avoid system names like AUX, COM1, CON, LPT3, NUL, PRN, TIME, ID when naming objects.

Databases and SD card

Problem	Possible cause	Solution
Error states "Disk IO error" at startup.	SD card has insufficient space.	Ensure the SD card has enough free space.
HMI panel stops the iX App and reboots repeatedly.	SD card removed while iX App runs.	Reconnect the SD card.
		Transfer a new project to the with the database location set to the **Hard drive**. See Change Location for the Active Databases.
Audit Trail database not included in file retention confirmation.	Project transferred with database set to SD card.	Remove SD card and delete the Audit Trail database via PC before transferring.

CPU performance

Problem	Possible cause	Solution
CPU load spikes when moving objects or hovering over buttons.	Windows hardware acceleration is on.	Turn off hardware acceleration in Windows.
	Graphics driver is outdated.	Install the latest graphics driver.
		Upgrade to a more powerful graphics card. See Recommended System Specifications for iX Developer.

Scripts

Problem	Possible cause	Solution
Navigating the script tree view is slow.	Project hasn't been saved lately.	Save the project.

TIP

For additional help, visit the Beijer Electronics forum, or contact the Beijer Electronics support.

Head Office
Beijer Electronics AB
Box 426
201 24 Malmö, Sweden
www.beijerelectronics.com / +46 40 358600

Beijer
ELECTRONICS