Beijer ELECTRONICS

iX Developer

MAEN832I, 2020-09

English

iX Developer User's Guide

Foreword

The iX Developer software is used to configure iX Panels and PC operated control applications, including applications for IPCs (Industrial PCs).

iX Developer makes it easy to create logical, flexible and effective HMI applications that provide the right information on the right occasion to operators and to other systems.

This manual is based on an example project that describes a step-by-step design of a project for iX Developer.

For more detailed information, please refer to the iX Developer help file.

Order no: MAEN832I

Contents

1 Introduction

The iX Developer software is used to configure iX Panels and PC operated control applications, including applications for IPCs (Industrial PCs) from Beijer Electronics.

iX Developer contains all basic functions needed in an application. The functions are tested and developed with customer needs and preferences in focus.

Pre-defined objects in iX Developer can be used to create complete process images, providing an overview of a complex application. You can customize the pre-defined objects or create objects of your own.

Communication drivers for a large number of controllers and automation equipment are available.

1.1 Controller

Operator panels can be connected to many types of automation equipment, such as PLCs, servos, and drives. Further on, the expression *controller* is used as a general term for the connected equipment.

1.1.1 Tags

Data values in a controller are referred to as *tags*. Tags may also 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.

1.2 Manual Structure

The USERS GUIDE is based on an example project that makes it easier to start using iX Developer. If the instructions in the example are followed carefully, this should result in a functional project that can be further developed, or used for inspiration. The target of the example is a PC, but all functions works similarly for all supported operator panels.

Detailed information about iX Developer is available in the help file, displayed by pressing **F1** while using the software.

The instructions in the USERS GUIDE are more detailed in the beginning. As the example progresses and you become familiar with iX Developer, instructions for tasks that are of a repetitive nature may be brief or omitted.

2 Starting iX Developer

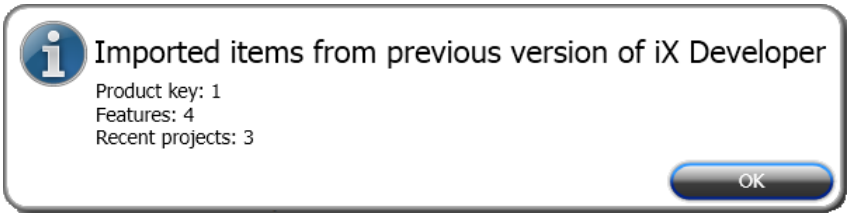
The installation creates an icon for the configuration tool in the group named iX Developer in Windows Start menu.

When starting a newly installed version for the first time the user gets the option to import the settings from the previous version.

This could also be done later from the File menu: Import settings

The following settings are imported:

- Product Key Features
- Recent Projects
- Added items in Component Library
- Additional Controls



After importing settings from previous version, information about what has been imported is displayed.

3 New Project

Objective:

- Creating a new project.
- Getting familiar with the tool windows and the layout of the desktop area.

3.1 Creating a New Project

1. Start iX Developer and select **Create New Project**.

   This starts a wizard that will guide you through creating the project.
2. Choose a **PC** with a **1024 × 768** resolution as target for the application. Click **Next**.
3. Select **DEMO** in the list of controllers. Click **Next**.

   The demo controller, including regular tags (data containers) and counters, is used to design and test a project directly on the development PC without connection to an external controller.
4. Give the project a name. For this tutorial use **DEMO_TEST**. Check that the suggested location is appropriate. If not, click **Browse** to select another location.
5. Click **Finish**.

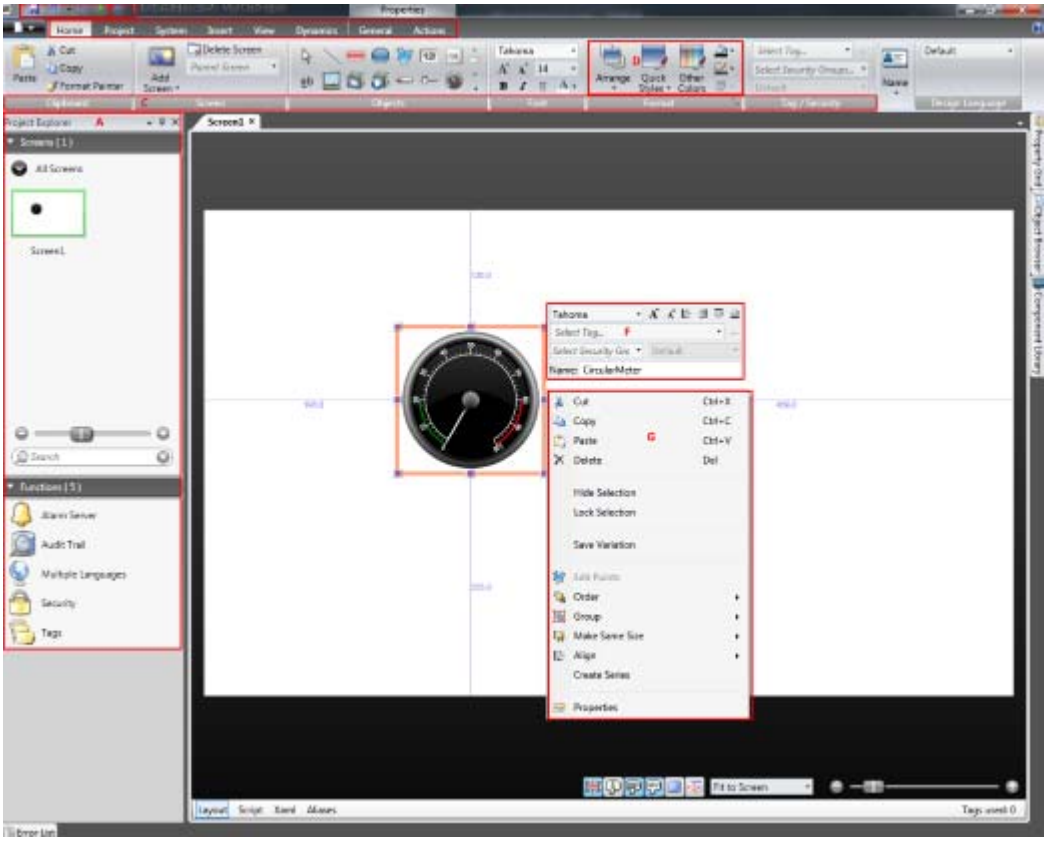
Project files can be stored anywhere in the computer environment where you have write access rights.

The project opens automatically.

3.2 Desktop Area

The desktop area displays screens and configuration pages for project components such as controllers and functions. The desktop area shows only one screen or component at a time. When multiple screens or components are opened a row of tabs are shown in the upper part of the desktop area. Clicking on a tab activates its contents for editing.

If there are more tabs open than can be displayed, navigation arrows in the upper part of the desktop area can be used to scroll between them.



Indication in picture	Desktop area component	Described in section
A	Project Explorer	*Project Explorer*
B	Ribbon tabs	*Ribbon Groups and Controls*
C	Control groups	
D	Controls	
E	Quick Access Toolbar	*Quick Access Toolbar*
F	Mini toolbar	*Mini Toolbar and Context Menu*
G	ContextMenu	

3.2.1 Project Explorer

When a new project is opened, an empty screen is active in the desktop area. The **Project Explorer** is docked to the left.

3.2.2 Ribbon Groups and Controls

Ribbon tabs are located in the top section of the tool window. Each ribbon tab holds one or several control groups. Each group contains a set of related controls. You use the controls to design screens, and to make settings for objects and controls in the project.

If you are not used to software with ribbon tabs, please spend a minute to get familiar with the ribbon concept.

3.2.3 Quick Access Toolbar

The **Quick Access** toolbar is always visible at the top of the desktop area. It contains the commands **Save**, **Undo**, **Redo**, **Run** and **Simulate** as iX Developer is started.

3.2.4 Mini Toolbar and Context Menu

When right-clicking objects in iX Developer, a mini toolbar and a context menu are displayed. The mini toolbar contains commands that are specific for iX Developer, to for example connect objects to controller tags. The context menu contains regular Microsoft application commands such as **Copy**, **Paste** etc.

4 Controller Tags

Objective:

- Defining a tag list for the project.
- Saving the project.

4.1 Adding Tags

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.

1. Click on **Tags** in the **Project Explorer**.
   The tags configuration page opens on the desktop. There is one element in the tags list by default.
2. Click on the first field (**Name**) in the first row.
   A cursor is placed next to the text **Tag1**.
3. Press [**TAB**] on the keyboard.
   The selection moves to the next field (**Data Type** under **Tags**). You do not need to change the data type.

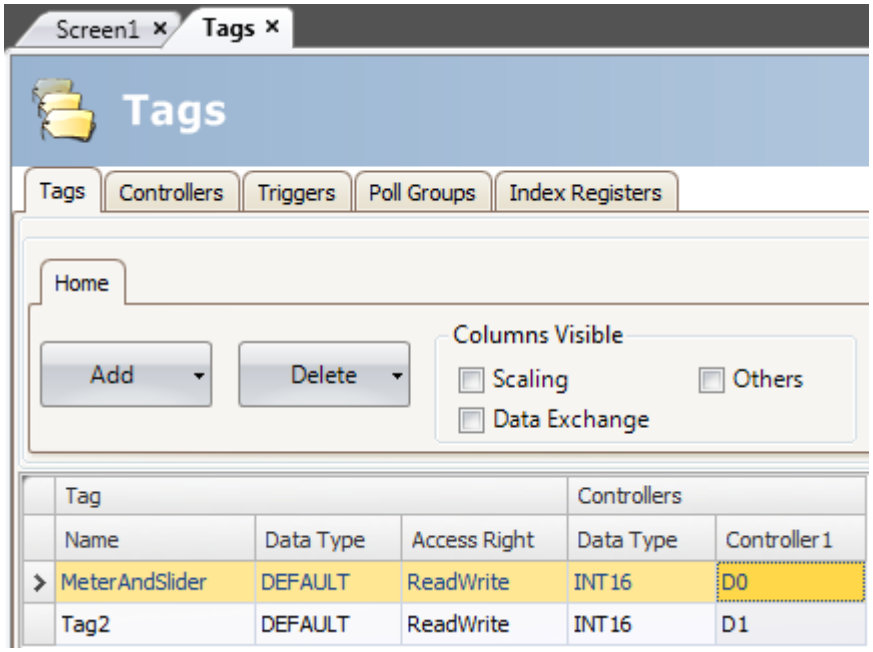
Note:
The data type for the tag can be used as a presentation format in some situations; for example to show the correct engineering unit when using scaling. **DEFAULT** follows the selection made for the **Data Type** under **Controllers**.

---

4. Press [**TAB**] on the keyboard again.
   The selection moves to the next field (**Access Right**). You do not need to change the access rights now.
5. Press [**TAB**] on the keyboard again.
   The selection moves to next field (Controller **Data Type** under **Controllers**). You do not need to change the data type now.
6. Press [**TAB**] on the keyboard again.
   The selection moves to next field (**Controller 1**).
7. Type **D0** in the **Controller 1** field.
   The entries in the **Controllers** column correspond to tags in the selected controller. There are predefined tags in the DEMO controller that can be accessed by their respective tag address, e.g. **D0** denotes an integer tag field.
8. Press [**TAB**] until the two first rows are completely filled in. Type **D1** for the second controller tag.
   Some fields are automatically filled and when needed, incremented. The data type is automatically changed depending on what you type in the **Controllers** column.

9. Rename **Tag1** to **MeterAndSlider**.

Note:
The **Name** of a tag is an identifier for the tag and can be any alphanumeric string, beginning with a letter (a-z, A-Z).



The **D0** tag will be used in the next section to control and observe a controller tag value in a screen.

4.2 Saving the Project

1. Click on the **Save** symbol in the **Quick Access Toolbar**.
   The project will be saved in the location that you selected when creating the project.

5 Editing Objects

Objective:

- Inserting a slider and a meter.
- Learning how to format and align objects.
- Testing the project; controlling and observing a controller tag value with objects in a screen.

5.1 Adding Objects

5.1.1 Meter

1. Click on the **Screen1** tab in the desktop area and make sure that the **Home** ribbon tab in the ribbon area is selected.
2. Select a circular meter from the **Objects** group. Place it somewhere in the upper left section of the screen.
3. Drag a corner handle to resize the meter to a suitable size where the meter needle and the scale are clearly visible.
4. Right-click on the meter and expand the **Select Tag...** dropdown list.
5. Select **MeterAndSlider** and then click **OK**.

5.1.2 Slider

1. Select a slider from the **Objects** group. You may need to expand the **Objects** group by clicking the small arrow at the lower right in order to select the slider.
2. Place the slider just below the circular meter on the screen.
3. Right-click on the slider and expand the **Select Tag...** dropdown list.
4. Select **MeterAndSlider** and then click **OK**.

5.1.3 Align

An object that is dragged snaps into position relative to other objects.

1. Slowly drag the slider up and down.

   Notice that the slider snaps into position at a short distance below the meter.
2. Slowly drag the slider to the left and to the right.

   Notice that the slider snaps into position and that snap lines appear when the slider is aligned with the meter.
3. Arrange the slider in a position closely below the meter and with its left edge aligned with the left edge of the meter.

5.1.4 Resize

1. Make a multiple selection of the two objects (point at an empty area in the screen and drag diagonally across the objects).

   A multiple selection (group) has one primary object. The primary object has an orange frame; the other objects have blue frames. When enforcing format commands on the group the primary object is used as a template.

   If the meter is not the primary object then:
2. Click on it to change the primary selection of the group to the meter.

   Now adjust the width of the objects in the group:
3. Click on the **Arrange** control, located in the **Format** group of the **Home** ribbon tab, and select **Make Same Width**.

5.1.5 Changing Appearance

1. Select the slider on **Screen1**.
2. Click on the **Quick Styles** control in the **Format** group and select a new color style.
3. Click the small arrow at the lower right of the **Format** group in order to make additional settings for outline, shadow/fill effects etc.
4. Select the meter on **Screen1**.
5. Select the **General** ribbon tab and locate the **Style** group. Try the different pre-defined styles and evaluate which style suits your preferences the best.

5.2 Running Project Test

The project can be compiled and run at almost any time. This allows you to test your design continuously in an iterative manner.

1. Click on the **Run** icon in the **Quick Access Toolbar**.

   The project is now validated, and when no errors are found, the project will be compiled and executes in the development environment.
2. Drag the slider handle back and forth.

   Since both objects are connected to the same tag, the meter needle follows as you change the value of the slider control.
3. Close the **Run** window.

6 Navigation and Screen Jumps

An iX Developer project consists of screens with objects, usually connected to controller tags. All screens have the same basic functions. A screen can be given specific properties to specialize its behavior in the project:

- **Startup Screen**: The startup screen is the first screen that is displayed in runtime. By default, **Screen1** is used as startup screen, but any screen can be designated startup screen by right-clicking on the screen and selecting **Set as Startup Screen**.
- **Background and Foreground Screen**: Any screen, except screens with aliases, can be used as a background or foreground screen by the other screens in the project. For more information about aliases, please refer to the iX Developer help file.
- **Screen Template**: A screen that is saved as a screen template can be used not only in the current iX Developer project, but also in all future projects.

Screen jumps are made with actions that can be assigned to e.g. buttons. When using the **Navigation Manager** to add screens and create links between screens, buttons are added automatically in the upper left corner of the screen that the link originates from.

Objective:

- Adding new screens and setting up screen jumps with buttons.

6.1 Screen Navigation

6.1.1 Navigation Manager

- Click on the **View** ribbon tab in the ribbon area. Click on **Navigation Manager**.

  The **Navigation Manager** opens in the desktop.

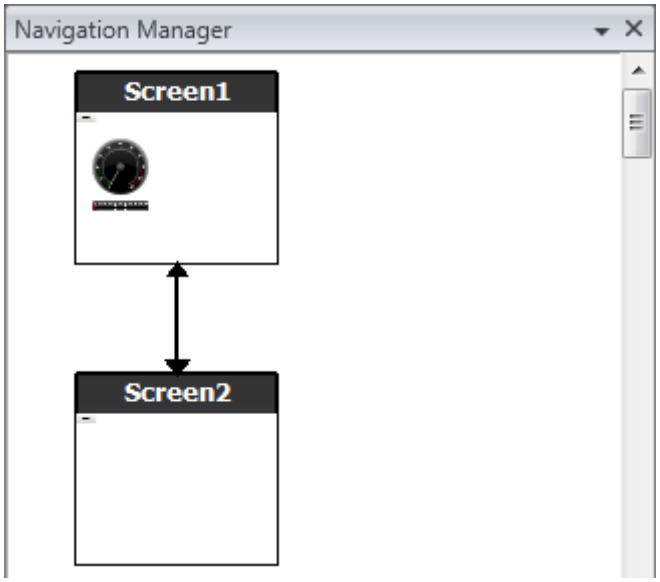
6.1.2 Adding a Screen

- Point at **Screen1**. Click and drag a connection from **Screen1** to anywhere in the **Navigation Manager** area.

  A new screen appears (**Screen2**). A button labeled **Screen2** appears in the upper left corner of **Screen1**.

6.1.3 Screen Jump

- Click and drag a connection from **Screen2** to **Screen1** in the **Navigation Manager**.
  A button labeled **Screen1** appears in the upper left corner of **Screen2**.



6.2 Background and Foreground Screen

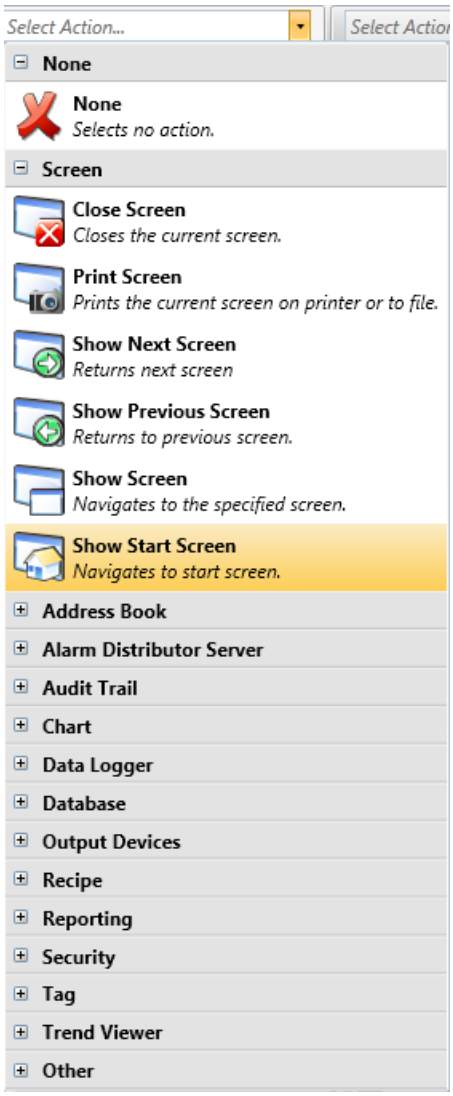
6.2.1 Adding a Screen

- Click on **Screen** on the **Insert** ribbon tab.
  A new screen (**Screen3**) is created in the project, and opens for editing.

6.2.2 Background and Foreground Screen

1. Make sure that **Screen3** is open for editing on the desktop.
2. Select a **Button** from the **Objects** group (located on the **Home** ribbon tab), and place it in the lower left area of **Screen3**. Label the button **Start Screen**.

3. Keep the button selected and click on the **Actions** ribbon tab. Select **Show Start Screen**, located in the **Screen** group, from the drop-down list for the **Click** action.



4. Open **Screen2** for editing by clicking on it in the **Project Explorer**.
5. Select the **Home** ribbon tab.
6. Open the **Parent Screen...** dropdown list.
7. Click **Screen3**.
8. Select **Background**.
9. Click **OK**.
10. Try to change location of the Start Screen button in **Screen2**. This is not possible. Notice that changes made to **Screen3** are reflected in **Screen2**.

There are now two navigation facilities from **Screen2** to **Screen1** (the start screen).

Setting a foreground screen is done by the same procedure, except that you select **Foreground** instead at step 8.

6.3 Running Screen Navigation Test

1. Run the project.
2. Verify that each of the buttons in **Screen2** performs a jump to **Screen1**.
   Since no screen has been set to be startup screen, **Screen1** remains startup screen for this project.
3. Close the **Run** window.

7 Trend Viewer

The trend viewer function stores register information from the controller in the operator panel. Real-time trend viewer as well as historical trend viewer is available.

Objective:

- Adding a trend viewer object with two curves.

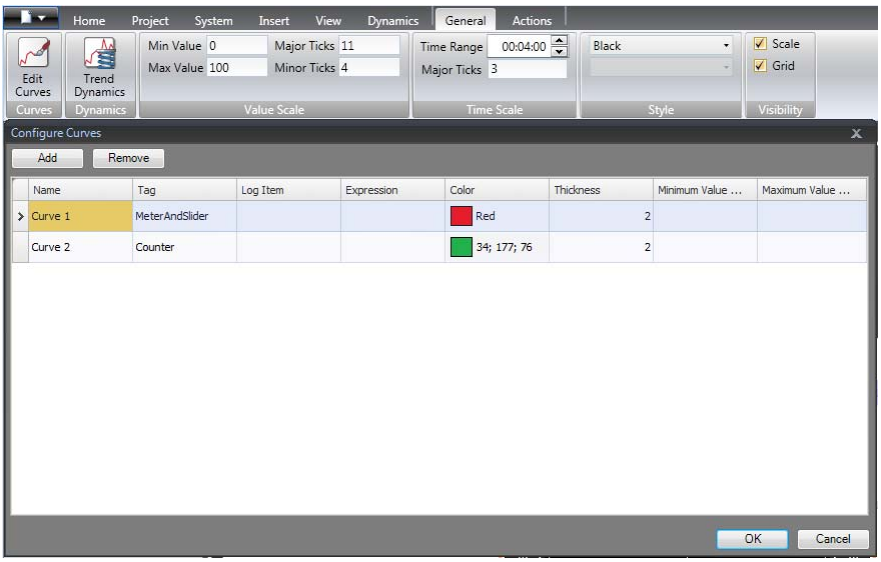
7.1 Adding a Real-Time Trend Viewer

1. Open **Screen1** for editing in the desktop.
2. Select the **Trend Viewer** object from the **Objects** group and place it on the screen.

7.1.1 Curves

1. Click on **Tags** in the **Project Explorer**.
2. Click **Add** to add a new tag.
3. Type **Counter** in the **Name** field and connect it to **C0** under **Controller1**.
   C0 is a counter that counts from 0 to 100 and back to 0 with a frequency of 1 Hz.
4. Open **Screen1**, make sure that the trend viewer object is selected, and click **Edit Curves** on the **General** ribbon tab.
5. Click **Add** to add a new curve.
6. Select **MeterAndSlider** in the **Tag** field, to connect the curve to the same tag that you used for the slider in **Screen1**.
7. Click **Add** to add a second curve.

8. Select **Counter** in the **Tag** field, and select another color for this curve.



9. Click **OK**.

7.2 Running Real-Time Trend Viewer Test

- Run the project and check that both curves are visible in the trend viewer. Test that **Curve 1** changes with the slider.

7.3 Trend Viewer History

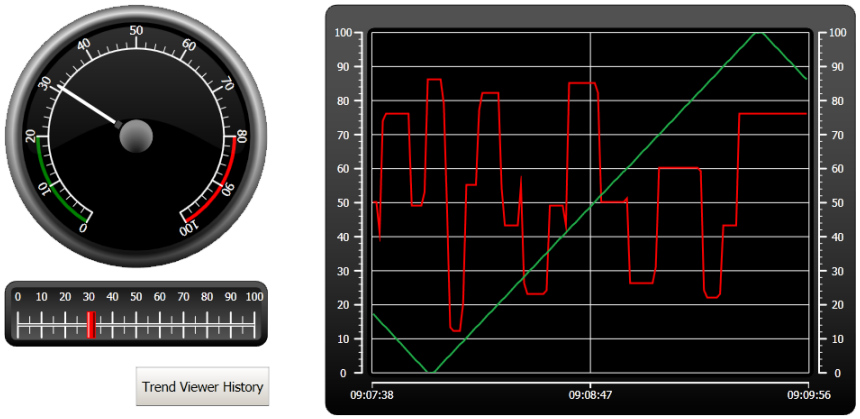
7.3.1 Actions

1. Open **Screen1** for editing in the desktop.
2. Click the **Home** ribbon tab.
3. Place a button to the left of the trend viewer object. Label the button **Trend Viewer History**.
4. Keep the button selected and click on the **Actions** ribbon tab.
5. Click the small arrow at the bottom right of the **Click** group.
6. Select **Click** in the **Trigger** dropdown list.
7. Click the **Select Action...** dropdown list, expand **Trend Viewer** and select **Trend Viewer History**.
8. Click the **Select Trend Viewer...** dropdown list and select **TrendViewer**.
9. Select **On** in the last dropdown list.
10. Click **OK**.
11. Select the trend viewer object and click on the **Actions** ribbon tab.
12. Click the small arrow at the bottom right of the **Mouse Down** group.

13. Select **Mouse Down** in the **Trigger** dropdown list.
14. Click the **Select Action...** dropdown list, expand **Trend Viewer** and select **Trend Viewer History**.
15. Click the **Select Trend Viewer...** dropdown list and select **TrendViewer**.
16. Select **Off** in the last dropdown list.
17. Click **OK**.

7.4 Running Historical Trend Viewer Test

1. Run the project.



2. Test that you can switch to the historical trend viewer with the **Trend Viewer History** button.
3. Go back to real-time trend viewer by clicking on the trend viewer object.

8 Alarm Management

Alarms are used to make the operator aware of events that require immediate action. An alarm is triggered when a certain condition is met. An alarm condition is designed as a logical evaluation of a tag value. Alarms can be divided into groups to create an order of priority.

Objective:

- Configuring the alarm list and designing an alarm object.

8.1 Alarm Indicator

When an alarm goes active, the alarm indicator becomes visible to alert the operator, regardless of which screen is active.

The appearance of the alarm indicator depends on the current alarm status.

Select **General** settings in alarm server properties page to decide for which alarm statuses to show the alarm indicator.

The alarm indicator will show the most severe alarm in the alarm list with the following indications:

- Flashing red when there is any active, unacknowledged alarm.
- Flashing green when no active alarms exist, but inactive unacknowledged alarms exist.
- Flashing green when there are only active acknowledged alarms.

The alarm indicator disappears when all alarms are both acknowledged and have returned to inactive status.

8.2 Alarm Server

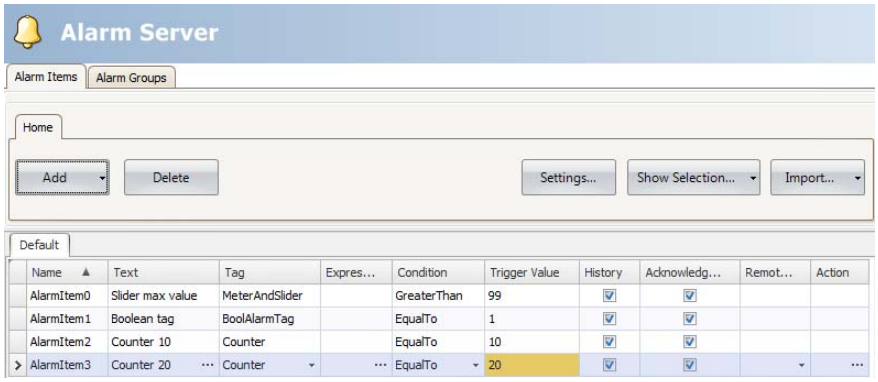
- Click on **Alarm Server** in the **Project Explorer** to open the **Alarm Server** configuration page.

8.2.1 Alarm Groups

The **Alarm Groups** tab is used to set up multiple alarm groups, e.g. when a project needs separate management of alarms for independent functions. In this example, the default alarm group is used.

8.2.2 Alarm Items

1. Click the **Alarm Items** tab.
2. Click **Add**.
   A new tag named **AlarmItem0** is added to the list.
3. Click the empty **Text** field and write **Slider max value**.
4. Select **MeterAndSlider** in the **Tag** field.
5. Select **GreaterThan** in the **Condition** field.
6. Change the value in the **Trigger Value** field to **99**.
7. Click **Add** again.
   A new tag named **AlarmItem1** is added to the list.
   This tag will be internal unless connected to a controller on the **Tags** configuration page, and using an internal tag works just fine for this example project. See chapter *Internal Tags* for information.
8. Click the empty **Text** field and write **Boolean tag**.
9. Click the **Tag** field and write **BoolAlarmTag**.
10. Select **EqualTo** in the **Condition** field.
11. Change the value in the **Trigger Value** field to **1**.
12. Add two more alarm items according to the image below.



13. Leave the columns for **History** and **Acknowledge Required** checked. Leave the other fields as they are.

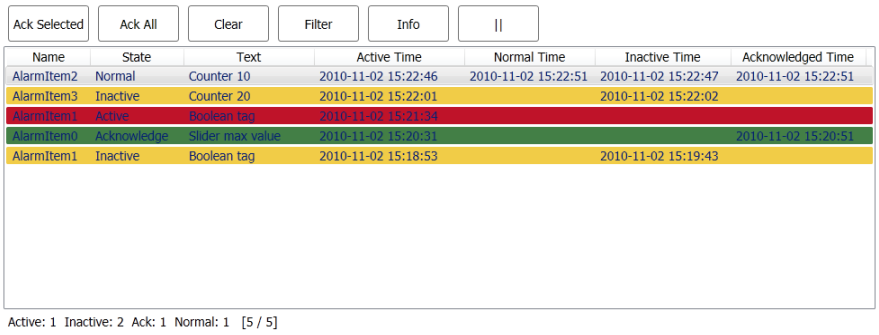
14. Make sure that all alarm tags can be controlled from the project screens or that they will be triggered by other mechanisms (the counter will trigger **AlarmItem2** and **AlarmItem3** after 10 and 20 seconds respectively).
15. Place a button to the left of the trend viewer object. Label the button **Set Alarm**.
16. Keep the button selected. On the **Actions** ribbon tab, select **Toggle Tag**, located in the **Tag** group, from the drop-down list for the **Click** action.
17. Select **BoolAlarmTag** in the **Select Tag** field.

8.3 Alarm Viewer

1. Open **Screen2** for editing in the desktop.
2. Click the small arrow at the bottom left of the **Objects** group and select **Alarm Viewer** under **HMI Controls**.
3. Click on the screen to place an alarm object on it.
   The columns and button placement can be customized in an alarm object.
4. Select the alarm object on the screen, and click on the **General** ribbon tab.
5. In the **Buttons** group, select **Top** in the **Position** dropdown list.
6. Adjust the size so that all button controls in the alarm object are visible.
7. Click on **Configure Columns** in the **Display Settings** group to customize the alarm information and the order of the columns in the alarm object.

8.4 Running Alarm Test

1. Run the project.
2. Test to trigger the alarms.



3. Click the **Ack All** button and observe the alarm indicator.
4. Press **Clear** to remove all alarms of normal status (acknowledged and inactive).

9 Recipes

Recipes are used to set or save a predefined group of tags in one operation.

Objective:

- Creating and using recipes to change multiple values.

9.1 Creating Recipe Tags

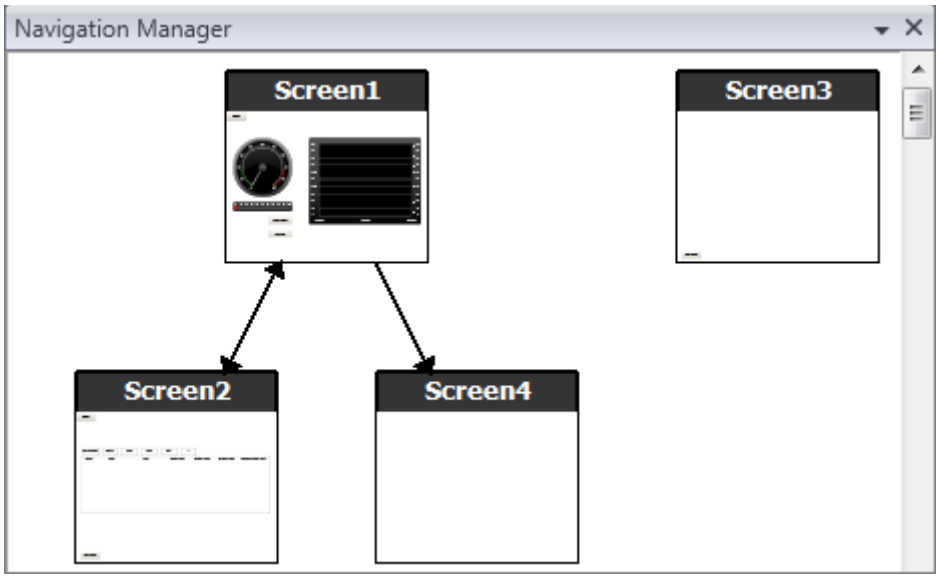
- Create a group of controller tags that the recipe should affect. Create three integer values according to the image below, for setting weight, length and width of an imaginary item.

	Tag			Controllers	
	Name	Data Type	Access Right	Data Type	Controller 1
	MeterAndSlider	DEFAULT	ReadWrite	INT16	D0
	Tag2	DEFAULT	ReadWrite	INT16	D1
	Counter	DEFAULT	ReadWrite	INT16	C0
	BoolAlarmTag	DEFAULT	ReadWrite	DEFAULT	
	Weight	DEFAULT	ReadWrite	INT16	D2
	Length	DEFAULT	ReadWrite	INT16	D3
I	Width	DEFAULT	ReadWrite	INT16	D4

9.1.1 Adding a Screen

1. Click on the **View** ribbon ftab in the ribbon area. Click on **Navigation Manager**.

2. In the **Navigation Manager**, point to **Screen1** and drag a connection to an empty spot in the screen navigation area.



A new screen (**Screen4**) is created in the project.

3. Open **Screen4** and select the **Home** ribbon tab.
4. In the **Screen** group, click the **Paren Screen** dropdown list.
5. Select **Screen3** and then select **Background**.
6. Click **OK**.
   This enables navigation from **Screen4** to **Screen1**.

9.1.2 Adjusting Navigation Buttons

- Open **Screen1**.
- Select the button labeled **Screen4** (in the upper left corner) and move it so the button placed under it (**Screen2**) becomes fully visible.

9.2 New Objects

1. Open **Screen4** for editing in the desktop.
2. Add a slider, found in the **Objects** group on the **Home** ribbon tab.
3. Right-click on the slider and open the **Select Tag...** dropdown list. to open the controller drop-down list.
4. Select **Weight** and click **OK**.

   This connects the **Weight** tag to the object.
5. Press **Ctrl**, then drag the slider across the screen to make a copy of it. Position the new slider and connect it to **Length**.
6. Repeat step 5, but connect the slider to **Width**.

9.2.1 Show Info

It is possible to show information about which tag each object is connected to, and if dynamics or actions are configured for the object. This is done by clicking on the **Show/Hide Info** button in the desktop area, or by using the keyboard shortcut **Ctrl + D**.

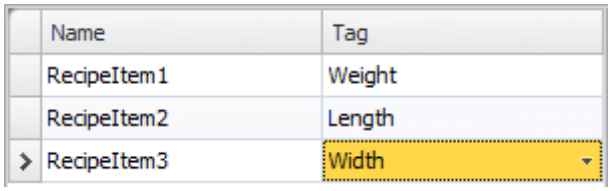
- Press **Ctrl + D** on the keyboard to check that the tags are correctly bound to the sliders in the screen.

9.3 Recipe Items

1. Click on **Recipe** on the **Insert** ribbon tab.

   The recipe configuration page opens in the desktop. The new recipe is also available from the **Project Explorer**.
2. On the **Tag Configuration** tab, click **Add** three times to add three new recipes.

3. Connect the items to tags according to the image below.



	Name	Tag
	RecipeItem1	Weight
	RecipeItem2	Length
>	RecipeItem3	Width

9.4 Saving a Recipe

1. Open **Screen4** for editing in the desktop.
2. Place a button next to the set of sliders for the recipe tags and label the button **Save Recipe**.
3. Keep the button selected and click on the **Actions** ribbon tab
4. Open the **Select Action...** dropdown list for the **Click** action.
5. Select **Save Recipe**, located in the **Recipe** group.
6. Make sure that **Recipe1** is selected in the next dropdown list.
7. Leave **Select Recipe data (optional)...** empty.

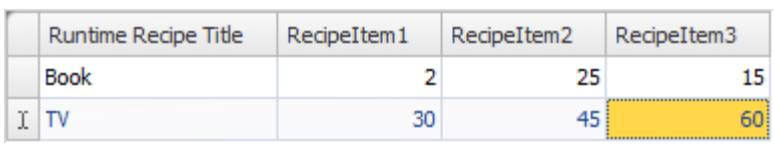
9.5 Loading a Recipe

1. Open **Screen4** for editing in the desktop.
2. Press **Ctrl**, then drag the **Save Recipe** button across the screen to make a copy of it.
3. Change the label to **Load Recipe**.
4. Keep the button selected and click on the **Actions** ribbon tab
5. Open the **Select Action...** dropdown list for the **Click** action.
6. Select **Load Recipe**, located in the **Recipe** group.
7. Leave **Select Recipe data (optional)...** empty.

9.6 Recipe Data

Create a predefined recipe by defining values on the **Runtime Data** tab of the recipe configuration page.

1. Open the recipe configuration page by clicking on **Recipe1** in the **Project Explorer**.
2. Click on the **Runtime Data** tab.
3. Click the **Add** button twice.
4. Change the values of the recipe items according to the image below.



	Runtime Recipe Title	RecipeItem1	RecipeItem2	RecipeItem3
	Book	2	25	15
I	TV	30	45	60

5. Open **Screen4** for editing.
6. Place a new button next to the set of sliders. Label the button **Load Book**.
7. Keep the button selected and click on the **Actions** ribbon tab
8. Open the **Select Action...** dropdown list for the **Click** action.
9. Select **Load Recipe**, located in the **Recipe** group.
10. Open the **Select Recipe data (optional)...** dropdown list and select **Book**.

9.7 Running Recipe Test

1. Run the project.
2. Test to set the sliders to various values and save the values in recipes.
3. Test to load the recipes.
   Check that the sliders change according to the recipe values.

10 Dynamics

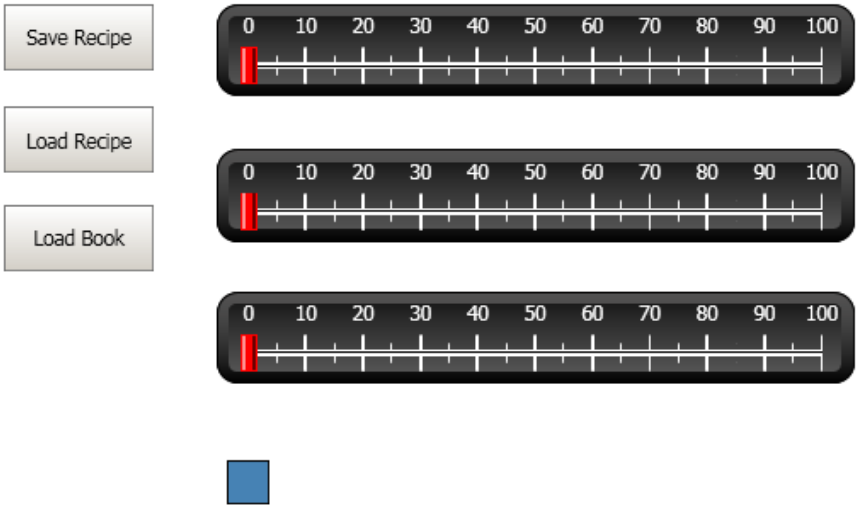
Dynamic object properties are used to move and resize objects based on controller tag values.

Objective:

- Changing size and color of an object based on tag value changes.

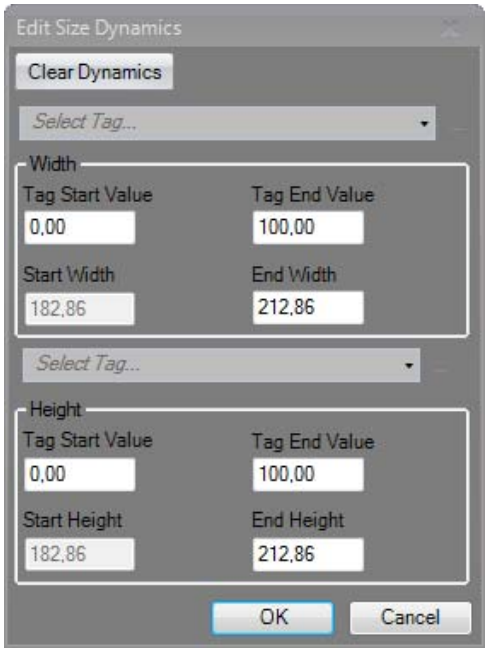
10.1 Creating an Object

1. Open **Screen4** for editing.
2. On the **Home** ribbon tab, select the red rectangle which is located in the **Objects** group.
3. Draw a small square below the set of sliders.



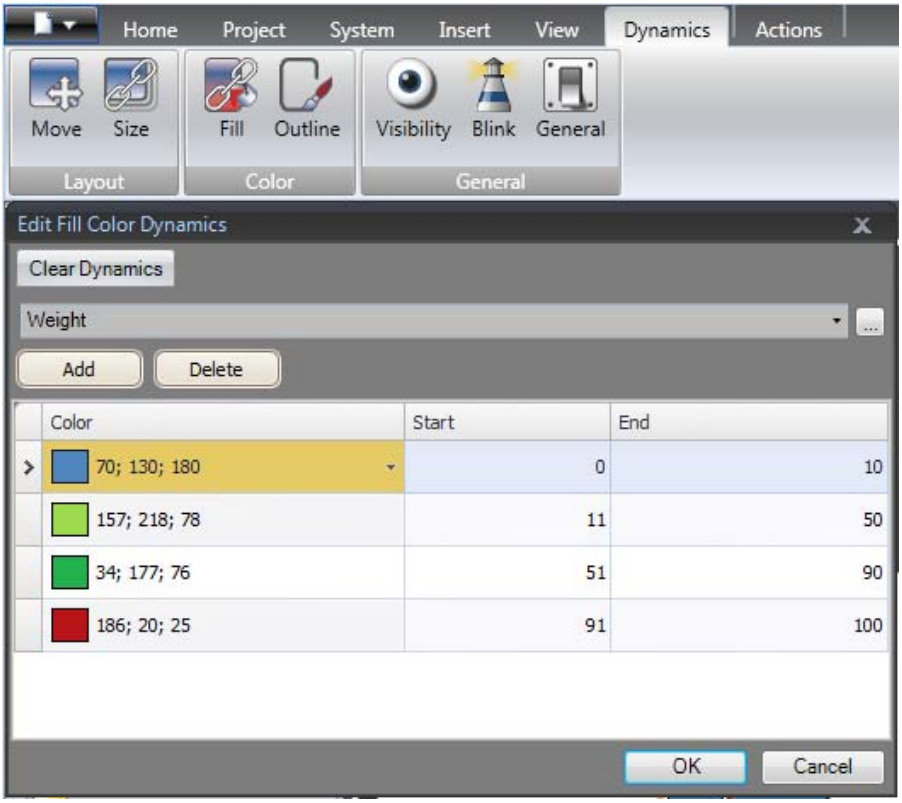
10.2 Resizing an Object

1. Select the square.
2. On the **Dynamics** ribbon tab, click on **Size** which is located in the **Layout** group.
3. Select the **Width** tag for **Width** and the **Length** tag for **Height**.
4. Click **OK**.
5. Adjust the enlarged size of the square directly on the screen and note the change of values in the **Edit Size Dynamics** window.



10.3 Coloring an Object

1. Select the square.
2. On the **Dynamics** ribbon tab, click on **Fill** which is located in the **Color** group.
3. Assign the **Weight** tag in the **Select Tag...** drop-down list.
4. Click **Add** four times.
5. Adjust the tag values to change the color of the square depending on the **Weight** tag value. The example in the picture below uses fill color in combination with a gradient.



6. Click **OK**.

10.4 Running Dynamics Test

1. Run the project.
2. Test to change the tag values with the sliders and by loading recipes. Observe what happens with the size and color of the small square.

11 Script

Scripts are used to manage functionality for objects. Scripts are written in C#.

Objective:

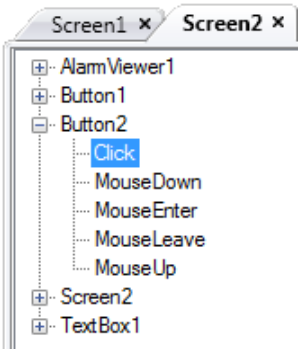
- Inserting a button and text box.
- Writing a script for the button to affect the text in the text box.

11.1 Adding Objects

1. Open **Screen2** for editing.
2. Click the **Home** ribbon tab.
3. Click the small arrow at the bottom right of the **Objects** group.
4. Select **Text Box** under **Windows Controls**.
5. Draw a **TextBox** on the screen.
6. Place a button on the screen and label it **Write Test**.

11.2 Script Tab

1. Select the **Write Test** button.
2. Click on the **Script** tab located in the lower left the desktop area.
   The view mode changes from Layout to Script.
3. Expand the **Button2** node.



4. Double-click on the **Click** node.
   You can now start typing script code for the **Click** event for **Button2**.
   A context sensitive name completion feature (IntelliSense) can be activated during typing with Ctrl + [Spacebar] and it triggers automatically when a period (‘.’) is typed after a code element.

5. Type the following as the click event code:

```
TextBox1.Text= "Test";
```

C# code on the Script tab:

```
public partial class Screen2
{

    void Button2_Click(System.Object sender, System.EventArgs e)
    {
        TextBox1.Text= "Test";
    }
}
```

11.3 Running Script

1. Run the project.
2. Click on the **Write Test** button and check that the text string assigned with the script code now appears in the text box.

12 Internal Tags

Internal tags can be used to calculate values that need not be represented in the controller, for example information only for the operator.

Objective:

- Writing a script to perform a calculation of the area using the length and width tags.
- Showing the result with an internal tag.

12.1 Adding Internal Tags

- Click on **Tags** in the **Project Explorer**.
  The tags configuration page opens on the desktop.

12.1.1 Area Tag

1. Add a tag and label it **Area**.
2. Change the data type to **FLOAT**.

12.1.2 Calculation Tag

1. Add a tag and label it **Calc**.
2. Change the data type to **BIT**.

	Tag			Controllers	
	Name	Data Type	Access Right	Data Type	Controller1
	MeterAndSlider	DEFAULT	ReadWrite	INT16	D0
	Tag2	DEFAULT	ReadWrite	INT16	D1
	Counter	DEFAULT	ReadWrite	INT16	C0
	BoolAlarmTag	DEFAULT	ReadWrite	DEFAULT	
	Weight	DEFAULT	ReadWrite	INT16	D2
	Length	DEFAULT	ReadWrite	INT16	D3
	Width	DEFAULT	ReadWrite	INT16	D4
	Area	FLOAT	ReadWrite	DEFAULT	
I	Calc	BIT	ReadWrite	DEFAULT	

Leaving the **Controller1** column empty keeps the tag internal, not connected to a controller.

3. Switch to **Script** view mode.
4. Expand the **Calc** node.

5. Double-click **ValueChange**.
   To access data and methods in C# control code the keyword "Globals" is used. The example uses explicit type casting ("(double)"), which is necessary for an overloaded operator (multiplication).
6. Calculate the area in the **ValueChange** node:

```
Globals.Tags.Area.Value =
    (double) Globals.Tags.Length.Value *
    (double) Globals.Tags.Width.Value / 100;
```

12.2 Creating an Analog Numeric

1. Open **Screen4** for editing.
2. Click the **Home** ribbon tab.
3. Select the **Analog Numeric** object in the **Objects** group.
4. Draw an analog numeric object below the set of sliders, clear from the rectangle object.
5. Right-click on the analog numeric object and connect it to the **Area** tag.
6. Keep the analog numeric object selected and click the **General** ribbon tab.
7. In the **Display Format** group, select **Decimal** in the dropdown list.
8. Set **Number of Decimals** to **2**.

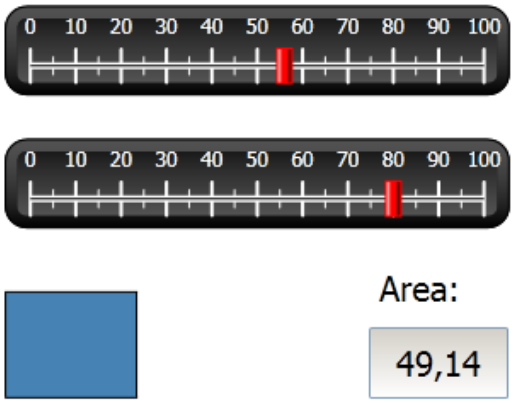
9. Select a **Text** object from the **Objects** group, and place a text object on the screen.
10. Place a text object on the screen in connection to the analog numeric object, and name it **Area:**.

12.2.1 Calculation Trigger

1. Open **Screen4** for editing.
2. Select the rectangle object.
3. Go to the **Actions** ribbon tab.
4. In the **Mouse Down** group, open the **Select Action...** dropdown list.
5. Expand the **Tag** node and select **Toggle Tag**.
6. Open the **Select Tag...** dropdown list and select the **Calc** tag.

12.3 Running Internal Tags Test

1. Run the project.
2. Test to set the sliders to various values. Click on the dynamic rectangle area and observe the change of the analog numeric control.



13 Object Browser

An overview of all objects included in a screen can be displayed in the **Object Browser**.

Objective:

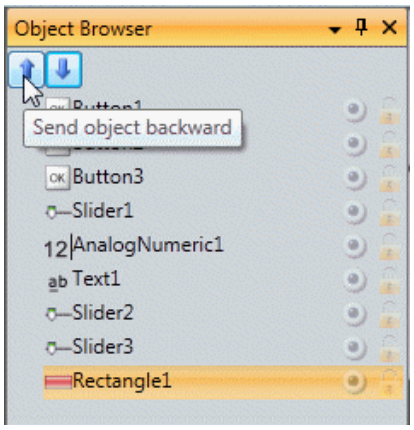
- Positioning objects in depth
- Locking objects
- Making objects invisible

13.1 Adding a Graphical Element

1. Open **Screen4** for editing.
2. Click the **Home** ribbon tab.
3. Select the **Rectangle** from the **Objects** group.
4. Draw a rectangle to fit as background of the group of sliders and buttons.
   The rectangle now totally obscures the other objects.

13.2 Using the Object Browser

1. Select **Object Browser** from the **View** ribbon tab.
2. Select the rectangle on the screen.
3. Click the **Send object backward** arrow button until all the buttons and sliders are visible.



4. Click the **Lock** icon next to the selected rectangle object in the **Object Browser**.
5. Try to move the rectangle on the screen by dragging it.
   The object is locked and cannot be moved. It cannot be selected in any way.

6. Select one of the buttons and click the **Visibility** button.
   The button is hidden. But if the project is run in iX Runtime, the object will show as normal.
7. Click the **Lock** button for the rectangle again and delete the rectangle.

14 Multiple Texts

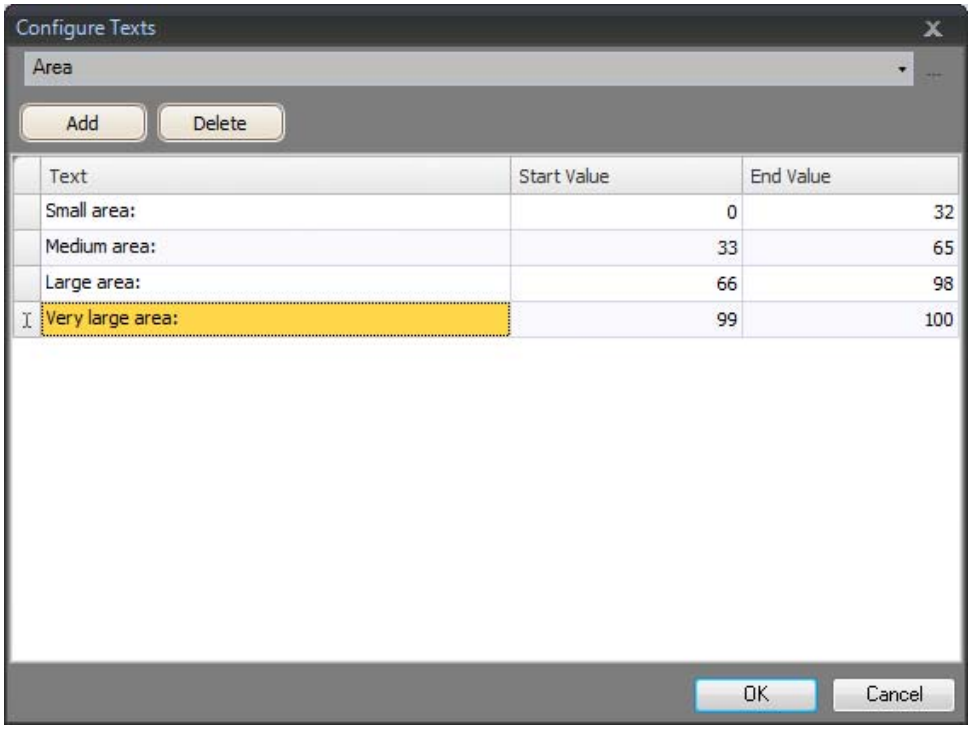
Text objects can be used to display textual information, and can also reflect changes in controller tags.

Objective:

- Presenting a variant text message that reflects the changes of the calculated area.

14.1 Configuring Texts

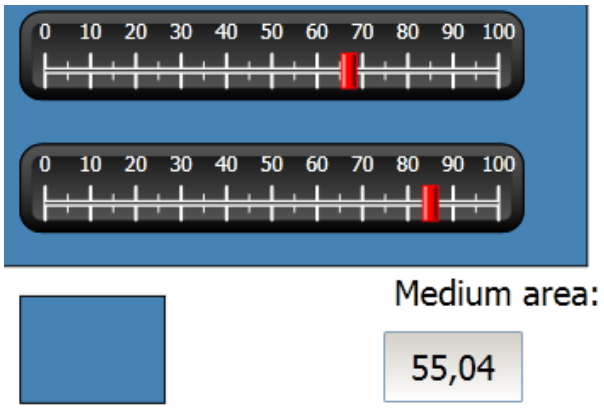
1. Open **Screen4** for editing.
2. Select the text object labeled **Area:**.
3. Click on the **General** ribbon tab.
4. Click on **Configure Texts** in the **Text** group.
5. Connect the text to the **Area** tag.
6. Add text strings and edit the intervals according to the example below.



With the default setting for the text object, **Autosize**, there is no need to adjust the object in order to make the longest string fit in runtime.

14.2 Running Multiple Texts Test

1. Run the project.
2. Test to set the sliders to various values. Click on the dynamic rectangle and observe the change of the analog numeric control. Check that the text is updated as well.



15 Security

Access to objects and actions in the project can be limited using security groups and user passwords.

Objective:

- Adding user names and setting passwords.
- Setting up login and logout control.
- Restricting access of recipe handling.

15.1 Security Configuration

- Click on **Security** in the **Project Explorer** to open the configuration page.

15.1.1 Security Groups

Security is handled by dividing users into security groups. These are configured on the **Groups** tab. In this example, the two default security groups, **Administrators** and **Operators**, are used.

15.1.2 Users

1. Select the **Users** tab on the **Security** configuration page.
2. Add a user and name it **SuperUser**.
3. Connect **SuperUser** to both **Administrators** and **Operators** under **Groups**.
4. Add another user, name it **User1** and connect it to the **Operators** group.
5. Enter passwords for the users.

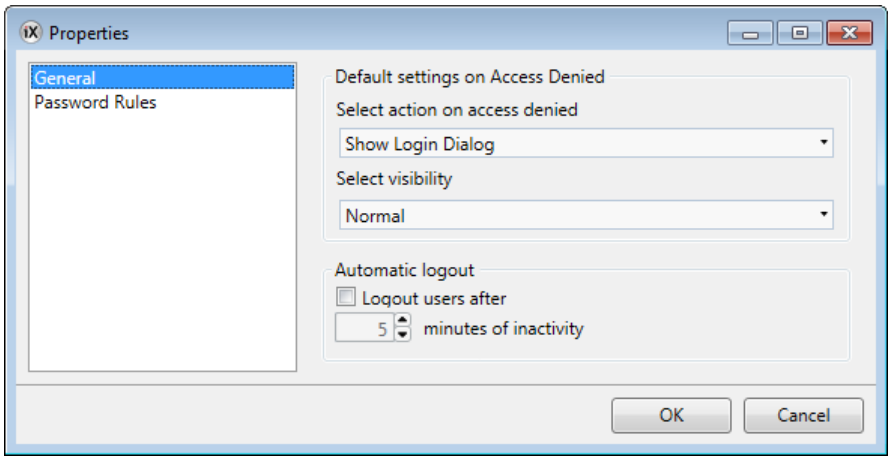
	Username	Password	Description	Groups
	Administrator	********		Administrators
	SuperUser	********		Administrators, Operators
I	User 1	oxpy		Operators

Note:
The password is converted to asterisks as you leave the password input cell.

15.2 Login Behavior on Access Denied

A login dialog can be displayed whenever a user tries to access an object that is restricted for the user group that the user belongs to.

1. Select the **Users** tab, and click the **Settings** button.
2. Select **Show Login Dialog** for action on access denied, and **Normal** for visibility.



3. Click **OK**.

15.3 Creating a Logout Button

1. Open **Screen3** (the background screen) for editing.
2. Place a button next to the **Start Screen** button. Label the button **Logout**.
3. Keeping the button selected and click the **Actions** ribbon tab.
4. Open the **Select Action...** dropdown list in the **Click** group.
5. Expand the **Security** node and select **Logout**.

15.4 Object Security

1. Open **Screen4** for editing.

15.4.1 Administrators

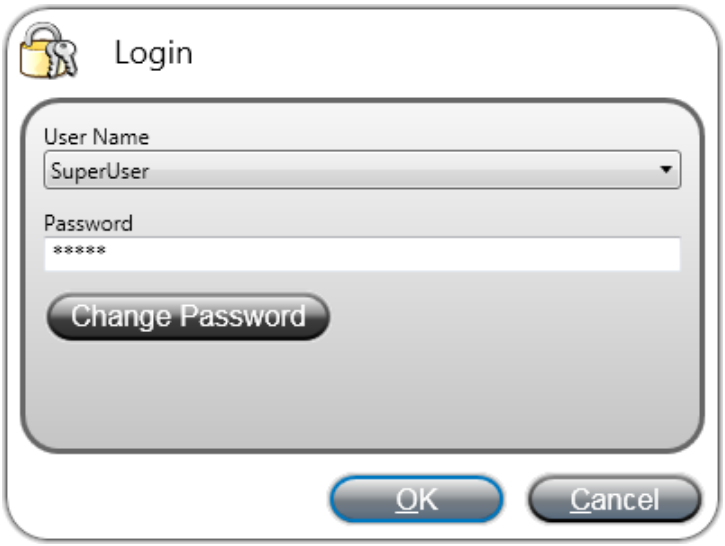
1. Right-click on the **Save Recipe** button and select **Administrators** for **Select Security Groups**.

15.4.2 Users

1. Right-click on the **Load Recipe** button and select **Operators** for **Select Security Groups**.

15.5 Running Security Test

1. Run the project.
2. Click the **Screen4** button.
3. Test to make sure that it is no longer possible to load or save recipes without logging in, and that the login dialog opens when any of the buttons are pressed.
4. Login as **Administrator** and save a recipe.
5. Test to load a recipe.
   The login dialog opens.
6. Login as **User1** and load a recipe.
7. Test to save a recipe.
   The login dialog opens.
8. Login as **SuperUser**. Test to save and load recipes.



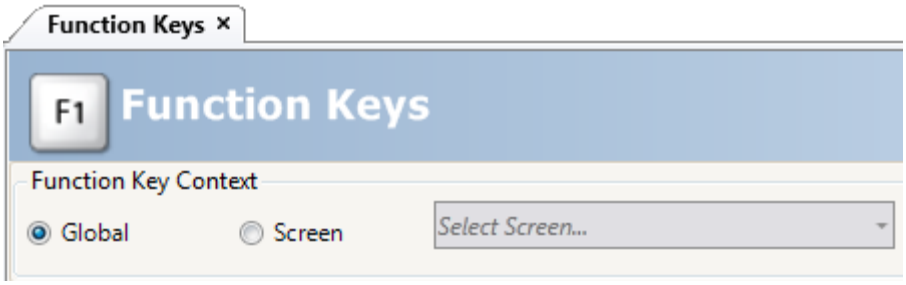
9. Log out.

10. Test that it is no longer possible to load or save recipes.

16 Function Keys

Function keys can be used to perform actions and execute scripts. This allows operator control of data and screen functionality independent of which screen is active.

Function keys can also be configured as local function keys, which means that they are applicable individual screens. In this example, global function keys are used.



Objective:

- Programming actions for function keys to change screen, set controller tag values, recipe management and display the login dialog.
- Programming function keys to execute scripts.

16.1 Defining Function Key Actions

1. Click on **Function Keys** in the **Insert** ribbon tab.

16.1.1 Show Screen

1. Click on ... under **Action** on the row for function key **F1** to open the **Properties** dialog.
2. Click **Select Trigger...** and select **Key Down**.
3. Expand the **Screen** node and select **Show Start Screen**.
4. Click **OK**.
5. Click on the row for function key **F2**.
6. Click ... under **Actions**.
7. Click **Select Trigger...** and select **Key Down**.
8. Expand the **Screen** node and select **Show Screen**.
9. Select **Screen2** in the next dropdown list.
10. Click **OK**.
11. Click on the row for function key **F3**.
12. Click ... under **Actions**.
13. Click **Select Trigger...** and select **Key Down**.
14. Expand the **Screen** node and select **Show Screen**.
15. Select **Screen4** in the next dropdown list.

16. Click **OK**.

16.1.2 Security

1. Click on the row for function key **F4**.
2. Click **...** under **Actions**.
3. Click **Select Trigger...** and select **Key Down**.
4. Expand the **Security** node and select **Login**.
5. Click **OK**.

16.1.3 Setting Controller Tag Values

1. Click on the row for function key **F5**.
2. Click **...** under **Actions**.
3. Click **Select Trigger...** and select **Key Down**.
4. Expand the **Tag** node and select **Set Analog**.
5. Select the **Weight** tag from the **Select Tag...** dropdown list.
6. Specify the analog value **50**.
7. Click **OK**.
8. Repeat the steps above to set up function keys **F6** and **F7** to control the **Length** and **Width** tags.

16.1.4 Recipe

- Set up function key **F8** to load **Recipe1**, and function key **F9** to save **Recipe1**. Leave **Select Recipe data (optional)...** empty.

16.1.5 Setting Time Zone, Region and Daylight Saving

- Set up function keys **F10** to set time zone, region and daylight saving. The action is located in the **Other** group.

Function Key	Actions
F1	Show Start Screen
F2	Show Screen
F3	Show Screen
F4	Login
F5	Set Analog
F6	Set Analog
F7	Set Analog
F8	Load Recipe
F9	Save Recipe
F10	Set Time Zone, Region and Daylight Saving ...

16.2 Defining Function Key Scripts

Function keys can also be used to trigger scripts.

16.2.1 Area Calculation

Program a function key with the area calculation for the rectangle object:

1. Click on the row for function key **F11**.
2. Select **Script** view mode by clicking on **Script** at the bottom of the screen.
3. Expand the **F11** node, and double-click on its **KeyDown** node.
4. Calculate the area on the **KeyDown** event with this code:

```
Globals.Tags.Area.Value =
    (double) Globals.Tags.Length.Value *
    (double) Globals.Tags.Width.Value / 100;
```

Using a function key script eliminates the need for the separate trigger tag (**Calc**).

16.2.2 Acknowledge All Alarms

Program a function key that acknowledges all alarms:

1. Expand the **F12** node, and double-click on its **KeyDown** node.
2. Type the following **KeyDown** event code:

```
Globals.AlarmServer.Acknowledge();
```

C# code in the script tab:

```
public partial class FunctionKeys
{

    void F11_KeyDown(System.Object sender, System.EventArgs e)
    {
        Globals.Tags.Area.Value =
            (double) Globals.Tags.Longitud.Value *
            (double) Globals.Tags.Ancho.Value /100;
    }

    void F12_KeyDown(System.Object sender, System.EventArgs e)
    {
        Globals.AlarmServer.Acknowledge();
    }
}
```

16.3 Running Function Key Test

1. Run the project.
2. Test that the defined functions keys (on the PC keyboard) perform the defined actions.

17 Cross Reference

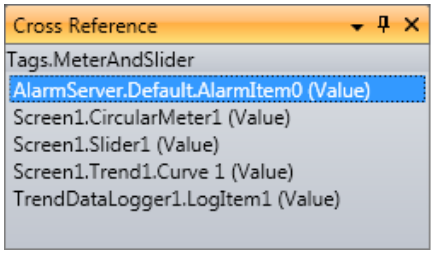
The **Cross Reference** tool provides an overview of where a specific tag is used in the current project.

Objective:

- Locating all occurrences of a specific tag quickly.

17.1 Using the Cross Reference Tool

1. Click **Tags** in the **Project Explorer**.
2. If you are in script view mode, then click **Design** at the bottom.
3. Select the **MeterAndSlider** row.
4. Click the **Cross Reference** button.
   The **Cross Reference** tool is displayed.



5. Double-click on **AlarmServer.Default.AlarmItem0 (value)** in the **Cross Reference** list.
   The **Alarm Server** configuration page opens on the desktop.
6. Double-click on **Screen1.Slider (Value)** in the **Cross Reference** list.
   Screen1 opens on the desktop, and **Slider1** is selected.

Head office

Beijer Electronics AB

Box 426

201 24 Malmö, Sweden

www.beijerelectronics.com / +46 40 358600