Kollmorgen Visualization Builder™

Quick Start Guide

Edition August 2010

Keep all manuals as a product component during the life span of the product.
Pass all manuals to future users / owners of the product.

KOLLMORGEN®

Because Motion Matters™

Record of Document Revisions

Revision	Remarks
8/31/2010	Preliminary edition

Preface

The Kollmorgen Visualization Builder (KVB) software is used to configure HMI panels and PC operated control applications, including applications for PACs (Programmable Automation Controller) from Kollmorgen.

The Kollmorgen Visualization Builder 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 Quick Start Guide is based on an example project that describes a step-by-step design of a project for Kollmorgen Visualization Builder.

For more detailed information, please refer to the Kollmorgen Visualization Builder User Manual.

IMPORTANT NOTICE

Disclaimer

The information in this manual was accurate and reliable at the time of its release. However, Kollmorgen. reserves the right to change the specifications of the product described in this manual without notice at any time.

This document contains proprietary and confidential information of Kollmorgen. The contents of the document may not be disclosed to third parties, translated, copied or duplicated in any form, in whole or in part, without the express written permission of Kollmorgen.

Registered Trademarks

Kollmorgen is a registered trademark of Danaher Corp.

Technical changes which improve the performance of the device may be made without prior notice! Printed in the Federal Republic of Germany.

August 2010

1 INTRODUCTION

The Kollmorgen Visualization Builder software (KVB) is used to configure HMI Panels and PC operated control applications, including applications for PACs (Programmable Automation Controller) from Kollmorgen.

KVB 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 KVB 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.

Note: When developing applications for a panel-based PAC or an independent Kollmorgen HMI panel, communication channels are automatically configured when KVB is launched through the Kollmorgen Automation Suite™ Integrated Development Environment (IDE).

1.1 Controller

Kollmorgen AKI series of HMI panels are optimized for use with Kollmorgen PACs. Ongoing use of the term controller implies any Kollmorgen PAC or variant.

1.1.1 Tags

Data values in a controller are referred to as tags. 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 Quick Start Guide is based on an example project that makes it easier to start using KVB. 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 KVB is available in the help file, displayed by pressing F1 while using the software.

The instructions in the Quick Start Guide are more detailed in the beginning. As the example progresses and you become familiar with KVB, instructions for tasks that are of a repetitive nature may be brief or omitted.

2 INSTALLATION

This sections contains installation and starting instructions for KVB and Kollmorgen Visualizer RT™.

2.1 KVB System Requirements

The following table provides the recommended system requirements.

KVB System Requirements	
RAM	2 GB
Processor	2 GHz or higher
Operating system	Microsoft Windows XP
Media player	Microsoft Media Player version 10 or later

2.2 Kollmorgen Visualizer RT System Requirements

The following table provides the recommended system requirements.

Kollmorgen Visualizer RT System Requirements	
RAM	1 GB
Processor	1.3 GHz or higher
Operating system	Microsoft Windows XP
Media player	Microsoft Media Player version 10 or later

Note: When user interface applications are running on a Kollmorgen panel-based PAC, ensure the USB hardware key, AKC-HMI-RK-xxx, is installed.

2.3 Starting KVB from the Kollmorgen Automation Suite

To start KVB:

1. Right-click on System in the Kollmorgen Automation Suite IDE Project Explorer.
2. Select Add HMI Device.
3. Right-click on the HMI Device to be added and select Add KVB Panel.
4. Double-click on the newly added Panel to start KVB.

For more information, see the Kollmorgen Automation Suite IDE Reference Manual.

2.4 Help

Help topics are provided when pressing **F1** while KVB is running.

3 NEW PROJECT

This chapter describes how to create a new project and introduces the tool windows, including the layout of the desktop area.

3.1 Creating a New Project (KVB Standalone)

To create a new Visualization Builder project from within KVB:

1. Launch KVB from the Windows Start menu:

   Start/Programs/Kollmorgen/Kollmorgen Visualization Builder/Kollmorgen Visualization Builder

2. Select **Create New Project**.

   A wizard displays to assist you in creating a new project.

3. Choose a **PC** with a **1024x768** resolution as target for the application. Click **Next**.

4. 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.

5. 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.

6. Click **Finish**.

   The project opens automatically.

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

   Note: When a separate Kollmorgen HMI panel is used in conjunction with your PAC, the Modbus/TCP address is loaded automatically if KVB is launched from the Kollmorgen Automation Suite IDE.

3.2 Creating a New Project (KAS IDE)

To create a new Visualization Builder project from within the KAS IDE:

1. Launch the KAS IDE from the Windows Start menu:

   Start/Programs/Kollmorgen/Kollmorgen Automation Suite/IDE

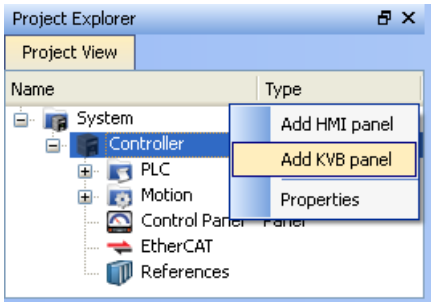
2. Click the **Create a New Project**  button or press **Ctrl+N**.

3. From the Controller Creation wizard, select the panel controller you would like to create a Visualization Builder project for and click **Next**.

4. From the Application Template dialog, choose the Motion Engine and click **Finish**.

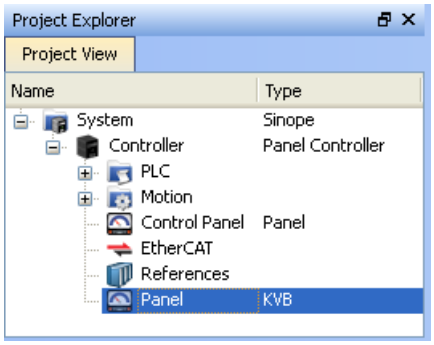
To Create a Visualization Builder project for a PAC:

1. From the KAS IDE Project Explorer, right-click on **Controller** and select **Add KVB Panel**.

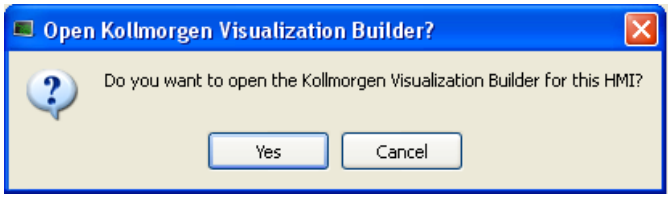


   The KVB panel is added for the PAC device in the Project Explorer.

2. Double-click on **Panel (KVB)** from the Project Explorer.



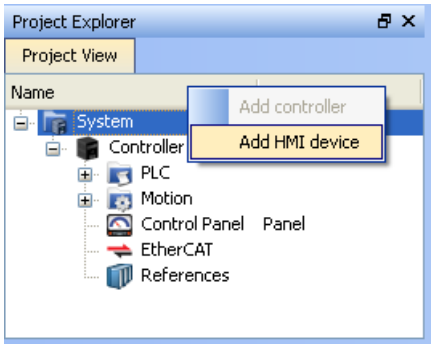
3. Select **Yes** to open with Kollmorgen Visualization Builder.



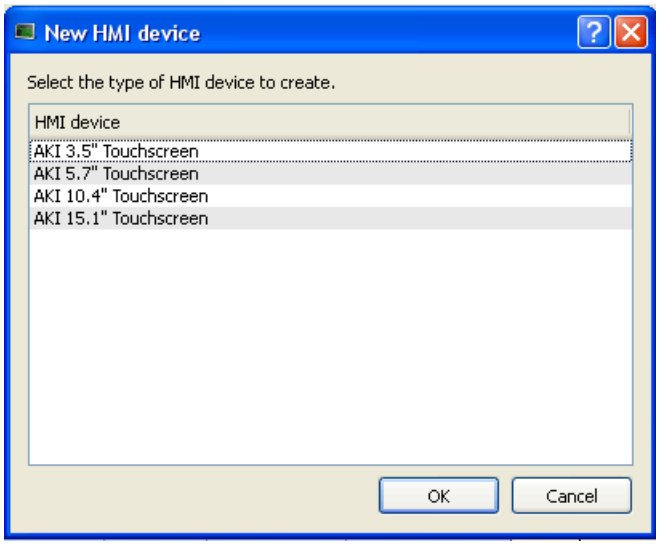
   The Kollmorgen Visualization Builder application launches and the PAC panel is now ready for design.

To create a Visualization Builder project for an HMI:

1. From the KAS IDE Project Explorer, right-click on **System** and select **Add HMI Device**.

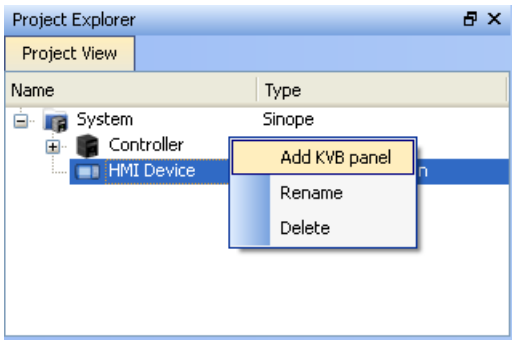


2. From the New HMI Device dialog, choose the HMI device you would like to create and click **Ok**.



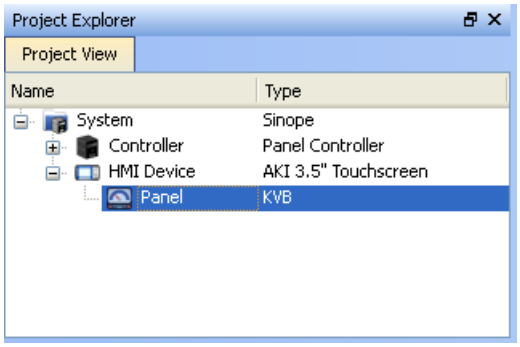
The HMI device is added in the Project Explorer.

3. From the Project Explorer, right-click on the HMI device and select **Add KVB Panel**.

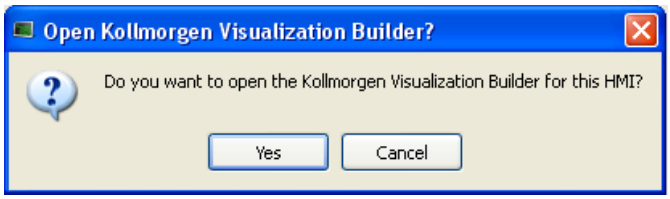


The KVB panel is added for the HMI device in the Project Explorer.

4. Double-click on **Panel (KVB)** from the Project Explorer.



5. Select **Yes** to open with Kollmorgen Visualization Builder.



The Kollmorgen Visualization Builder application launches and the HMI panel is now ready for design.

3.3 Desktop

This section describes the tool windows and desktop layout for the KVB.

3.3.1 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.

3.3.2 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.

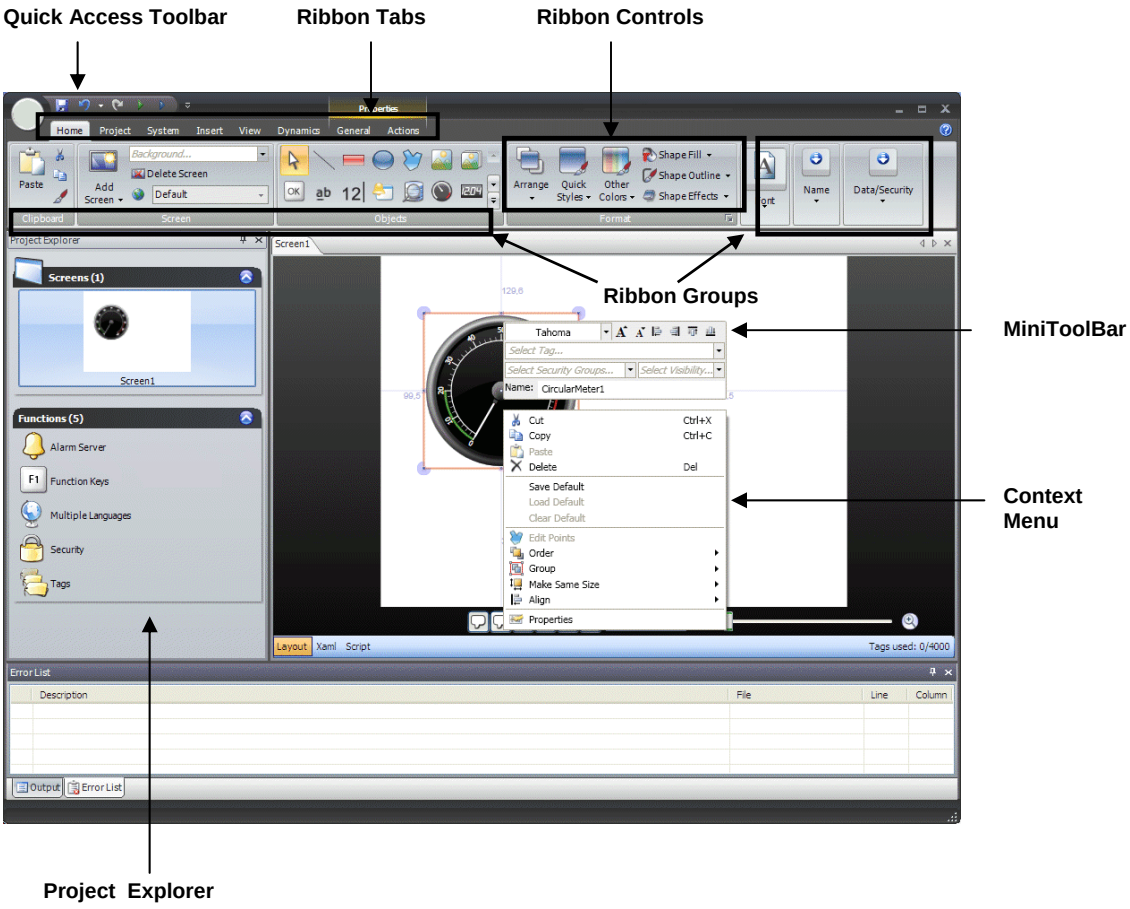
If you are not used to software with ribbon tabs, please spend a minute to get familiar with the ribbon concept.

3.3.3 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 KVB is started. The **Quick Access** toolbar can be customized to contain any other object from the ribbon tabs.

3.3.4 MiniToolBar and ContextMenu

When right-clicking objects in KVB, a **MiniToolBar** and a **ContextMenu** are displayed. The MiniToolBar contains commands that are specific for KVB, for example connecting objects to controller tags. The ContextMenu contains common Microsoft application commands such as **Copy, Paste** etc.



4 CONTROLLER TAGS

This chapter describes how to define a tag and save the project in KVB.

4.1 Importing Tags

When KVB is launched through the Kollmorgen Automation Suite IDE, all selected PLC variables are automatically imported.

4.2 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 are no elements in the tags list at this point.

2. Click on the first field (**Name**) in the first row.

   A highlighted row is added and the text **Tag1** displays.

3. Press [**TAB**] on the keyboard.

   The selection moves to next field (**Data Type**). You do not need to change the data type now.

4. Press [**TAB**] on the keyboard again.

   The selection moves to 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 1**).

6. Type **D0** in the **Controller 1** field.

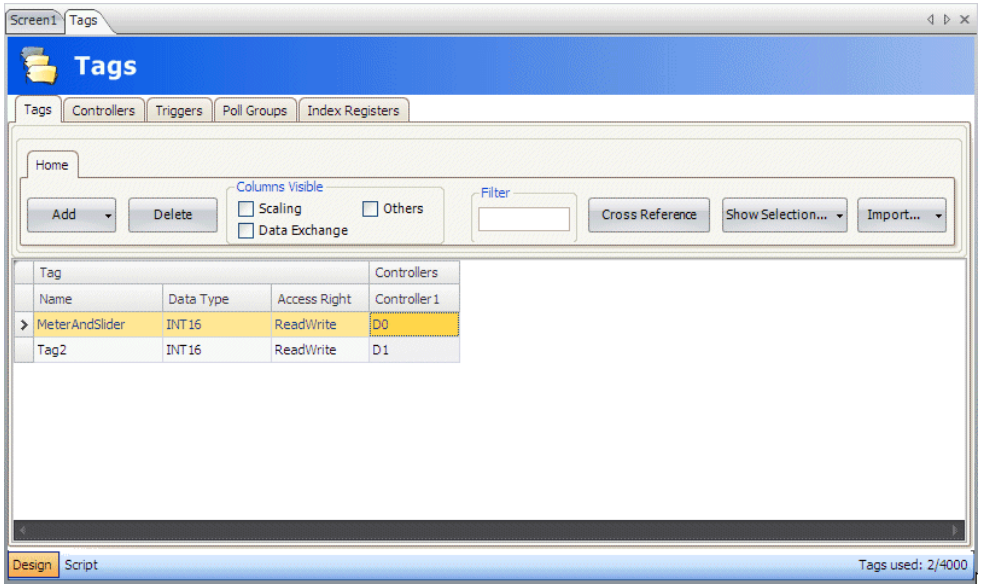
   The entries in the **Controllers** column correspond to tags in the selected controller. There are predefined tags in the DEMO controller that can be accessed by their respective tag address, e.g. **D0** denotes an integer tag field.

7. 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.

   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).

8. Rename Tag1 to **MeterAndSlider**.



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

4.3 Saving the Project (KVB Standalone)

Follow the procedure below to save the Visualization Builder project when using KVB in standalone mode.

1. Click on the **Save** symbol in the **Quick Access Toolbar**.

   The project is saved in the location that you selected when creating the project.

4.4 Saving the Project (KAS IDE)

Follow the procedure below to save the Visualization Builder project within KAS.

1. From the KVB **Quick Access Toolbar**, click on the **Save** symbol.

   Note: You do not need to save the Visualization Builder project separately from KAS.
2. Close KVB.
3. From the KAS IDE, click the **Save Project** button or press **Ctrl**+**S**.
4. From the **Project Save** dialog, choose a location to save your project and click **Save**.

   Your Visualization Builder and KAS project are now saved.

5 EDITING OBJECTS

This chapter describes how to insert a slider and meter, format and align objects, and test the project.

5.1 Adding Objects

Follow the procedure below to add an object such as a meter or slider.

5.1.1 Meter

To add a meter:

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

5.1.2 Slider

To add a 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. Place it just below the circular meter on the screen.
2. Right-click on the slider. Click on **Select Tag**... in the MiniToolBar. Select the **MeterAndSlider** tag by clicking on it in the drop-down menu, 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

To resize an object:

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** tab, and select **Make Same Width**.

5.1.5 Changing Appearance

To change the appearance of an object.

1. Select the slider on Screen1.
2. Click on the **Quick Styles** control in the **Format** group and 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 select **Styles** group. Try the different predefined styles and evaluate which style suits your preferences the best.

5.2 Running the Project

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.

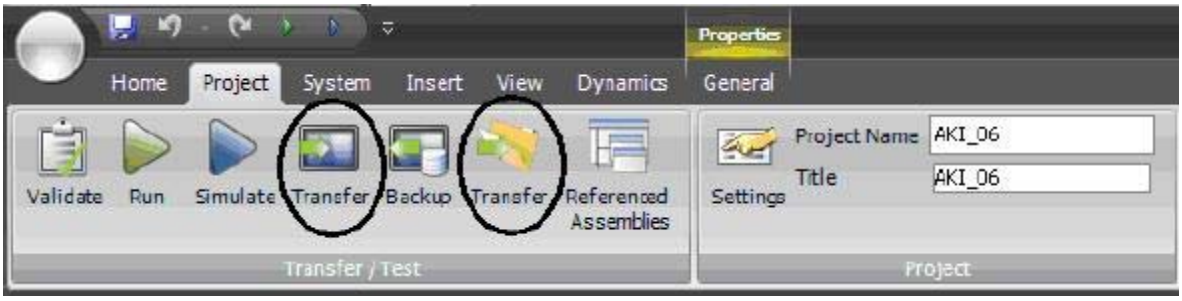
2. Drag the slider handle back and forth.

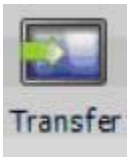
   Since both objects are connected to the same tag, the meter needle follows as you change the value of the slider control.

3. Close the **Run** window.

5.3 Transferring Project to Panel PAC and Standalone HMI

The following options are available for downloading a Kollmorgen Visualization Builder project to a Panel PAC or HMI panel:





Transfers the project via an Ethernet connection to an HMI panel.

Transfers the project via Program file (on memory stick) to an HMI panel or Panel PAC.

When transferring via memory stick, a project file folder is created. Inside the program file folder there is a **.exe** with the same name.

If loading to an HMI, insert the memory stick with the project file folder in the USB port. Then power up the HMI. A dialog box for storing the project file on the HMI memory will appear. After storing the project file on the HMI, on subsequent HMI startups the Kollmorgen Visualization Builder program will also startup.

If loading on a Panel PAC, insert the memory stick with the program file into the USB port on the Panel PAC. Transfer the program file to the PAC memory. If desired, the Kollmorgen Visualization Builder program can be executed on power up of the PAC by adding to the startup program list.

6 NAVIGATION AND SCREEN JUMPS

This chapter describes how to add new screens and setup screen jumps with buttons.

6.1 Screens

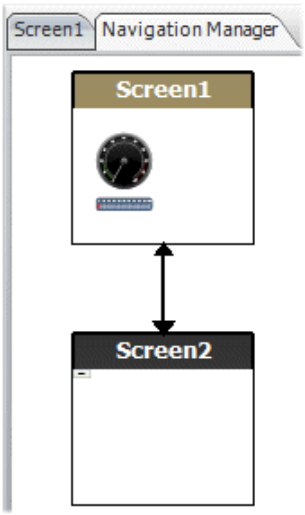
A Visualization Builder project consists of screens with objects, usually connected to controller tags. All screens have the same basic functions. A screen can be given specific properties to specialize its behavior in the project:

- **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 Screen —** Any screen can be used as a Background Screen by the other screens in the project.
- **Screen Template —** A screen that is saved as a Screen Template can be used in the current project and 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.

6.2 Screen Navigation

This section describes how to use the Navigation Manager, add a screen, and screen jumps.



6.2.1 Navigation Manager

1. Click on the **View** tab in the ribbon area. Click on **Navigation**.

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

6.2.2 Adding a Screen

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

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

6.2.3 Screen Jump

3. Click and drag a connection from **Screen2** to **Screen1** in the **Navigation Manager**.

   A button labeled Screen1 displays in the upper left corner of Screen2.

6.3 Background Screen

This section describes background screens.

6.3.1 Adding a Screen

To add a screen:

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

   A new screen (Screen3) is created in the project, and opens for editing.

6.3.2 Background Screen

Editing a background 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** 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** tab. Select **Show Start Screen** from the drop-down list in the **Click** group.
4. Open Screen 2 for editing by clicking on it in the **Project Explorer**.
5. Select the **Home** tab. Select **Screen3** in the **Background Screen** drop-down list in the **Screen** group.
6. 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).

6.4 Running Screen Navigation Test

To run a screen navigation test:

1. Run the project.
2. Verify that each of the buttons in Screen2 performs a jump to Screen1. Since no screen was setup as a Startup Screen, Screen1 remains Startup Screen for this project.
3. Close the **Run** window.

7 TREND

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

This chapter describes how to add a trend object with two curves.

7.1 Adding a Real-Time Trend

To add a real-time trend:

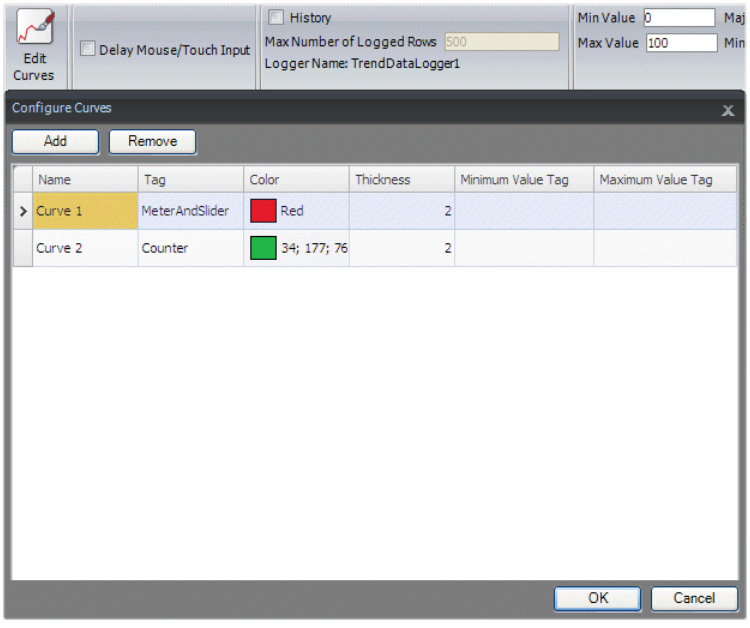
1. Open **Screen1** for editing in the desktop.
2. Select the **Trend** object from the **Objects** group and place it on the screen.

7.1.1 Curves

To add a curve:

1. Click on **Tags** in the Project Explorer and add a tag. Type **Counter** in the Name field and connect it to **C0** in Controller1.

   Note: C0 is a counter that counts from 0 to 100 and back to 0 with a frequency of 1 Hz.
2. Open Screen1, make sure that the trend object is selected, and click **Edit Curves** on the **General** tab.
3. Add a curve and connect it to the same tag that you used for the slider in Screen1.
4. Add a second curve and connect it to the Counter tag you just added, and select another color for this curve.



5. Click **OK**.

7.2 Running Real-Time Trend Test

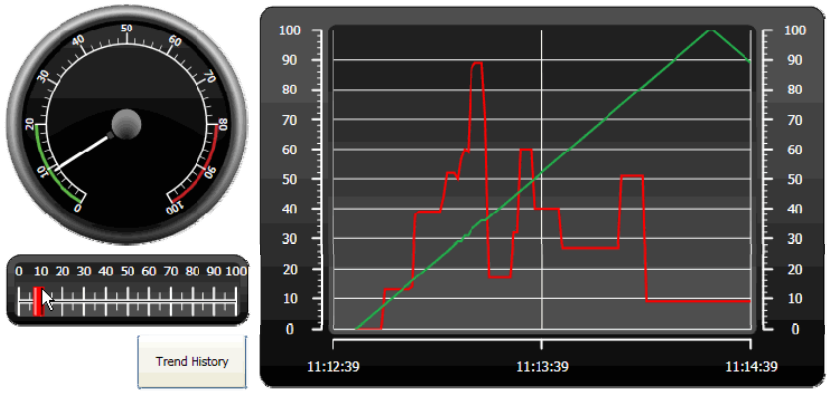
To run a real-time test:

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

7.3 Trend History

To set a trend history:

1. Open Screen1 for editing in the desktop.
2. Select the trend object and check the **History** box in the **Log Settings** group.
3. Place a button to the left of the trend object. Label the button **Trend History**.
4. Keep the button selected and click on the **Actions** tab. Select **Show Trend History** from the **Click** drop-down list. Select **Trend1** from the **Select Trend** dropdown list.
5. Select the trend object and click on the **Actions** tab. Select **End History Mode** from the **MouseDown** drop-down list. Select **Trend1** from the **Select Trend** drop-down list.



7.4 Running Historical Trend Test

To run a historical trend test:

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

8 ALARM MANAGEMENT

Alarms are used to make the operator aware of events that require immediate action. An alarm is set 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.

This chapter describes how to Configure the alarm list and design an alarm object.

8.1 Alarm Indicator

When an alarm goes active, the Alarm Indicator becomes visible to alert the operator. The appearance of the Alarm Indicator is configured on the **General** ribbon tab for the Alarm Server. The Alarm Indicator will show the most severe alarm in the alarm list with the following indications:

- Flashing red when there is any active, un-acknowledged alarm.
- Flashing yellow 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

To open the Alarm Server configuration page:

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

8.2.1 Alarm Groups

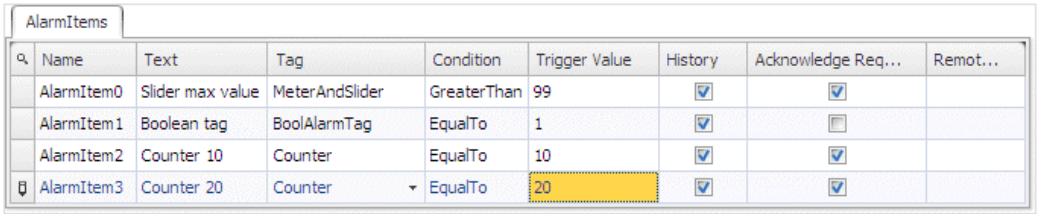
The Alarm Groups tab is used to set up multiple alarm groups, e.g. when a project needs separate management of alarms for independent functions.

The default alarm group is used in this example.

8.2.2 Alarm Items

To add an alarm item:

1. Select the **Alarm Items** tab. Add alarms based on tags in the controller tag list.
2. Define a digital tag (**M0**, named BoolAlarmTag) directly in the alarm list by clicking **Add** in the tag selection list. This tag will be internal unless connected to a controller on the Tags configuration page, and using an internal tag is appropriate for this example project. For more information about tags, see Chapter 12 Internal Tags.



Leave the columns for **Acknowledge Required** and **History** checked. Leave the field for **Remote Acknowledge** empty.

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).

3. Place a button to the left of the trend object. Label the button **Set Alarm**.
4. Keep the button selected. On the **Actions** tab, select **Toggle Tag** in the dropdown list of **Click** group. Select **BoolAlarmTag** in the **Select Tag** field.

8.3 Alarm Viewer

To add the alarm item to the screen:

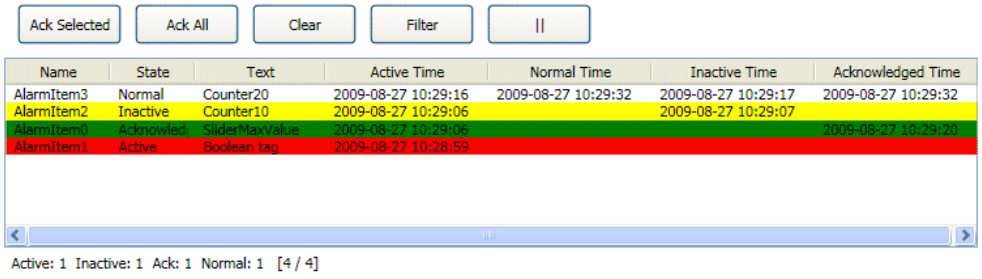
1. Open Screen2 for editing in the desktop.
2. Select the **AlarmViewer** from the **Objects** group and place it in the screen.

   The columns and button placement can be customized in an alarm object.
3. Select the alarm object on the screen, and click on the **General** tab. In the **Buttons** group, click on **Position** and select to place the buttons against the Top border.
4. Adjust the size so that all button controls in the alarm object are visible.
5. Use **Show Columns** in the **Settings** group to customize the alarm information and the order of the columns in the Alarm Viewer.

8.4 Running Alarm Test

To run an alarm test:

1. Run the project.
2. Test to trigger the alarms.
3. Press the **Ack All** button and observe the Alarm Indicator.



9 RECIPES

Recipes are used to set or save a predefined group of tags in one operation. This chapter describes how to create and use recipes to change multiple values.

9.1 Creating Recipe Tags

This section describes how to create a recipe tag.

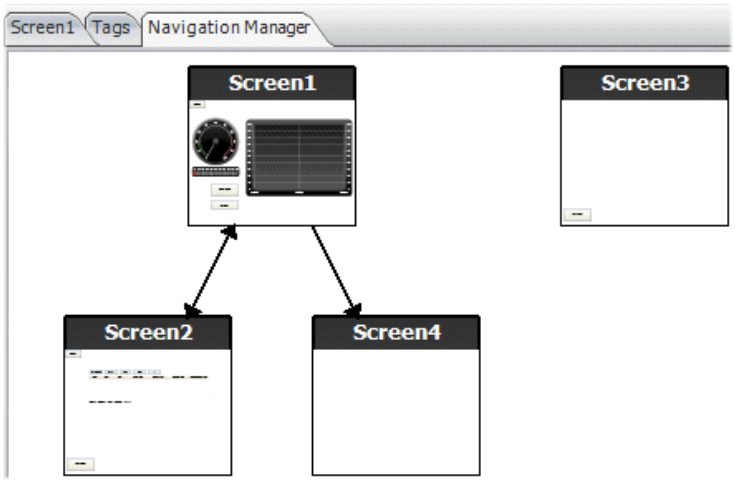
1. Create a group of controller tags that the recipe should affect. Use three integer values to set weight, length and width of an imaginary item.

Function Key	Action
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

9.1.1 Add a Screen

To add a screen:

1. Open 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.

2. Open Screen4 and select the **Home** tab. In the **Screen** group, select **Screen3** in the **Background Screen** drop-down list.

   This enables navigation from Screen4 to Screen1.

9.1.2 Adjusting Navigation Buttons

To adjust the navigation buttons:

1. 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

To modify a new object:

1. Open Screen4 for editing in the desktop, and add a slider.
2. Right-click on the slider and point to **Select Tag**... to open the controller dropdown list in the MiniToolBar. Select **Weight**.

   This connects the Weight tag to the object and the list closes.
3. Press **Ctrl**, then drag the slider across the screen to make a copy of it. Position the new slider and connect it to **Length**. Repeat for the **Width** tag.

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, by clicking on the **Show/Hide Info** button in the desktop area, or by using the keyboard shortcut **Ctrl + D**.

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

9.3 Recipe Items

To add a new recipe item:

1. Click on **Recipe** on the **Insert** tab to add a new recipe.

   The recipe configuration page opens in the desktop. The new recipe is also available from the **Project Explorer**.

2. Enter a group of tags to set in the recipe on the **Tag Configuration** tab.

Name	Tag
RecipeItem1	Weight
RecipeItem2	Length
RecipeItem3	Width

9.4 Saving a Recipe

To save a recipe:

1. Open Screen4 for editing in the desktop and place a button next to the set of sliders for the recipe tags. Label the button **Save Recipe**.
2. Click on the **Actions** tab and select Save **Recipe** from the drop-down list for the **Click** group. Make sure that **Recipe1** is selected.

   Note: Leave Recipe data empty.

9.5 Loading a Recipe

To load a recipe:

1. Open Screen4 for editing in the desktop, make a copy of the Save Recipe button.
2. Change the label to **Load Recipe**, and to load Recipe1 by selecting it for the **Click** action **Load Recipe**.

   Note: Leave Recipe data 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. Enter tag values to set in a recipe. Enter a name (**Runtime Recipe Title**) for the recipe.

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

3. Open Screen4 for editing. Place a new button next to the set of sliders. Label the button **Load Book**.

4. Select **Load Recipe** in the **Click** drop-down list.
5. Select **Recipe1**, and select **Book** for recipe data.

9.7 Running Recipe Test

To test a recipe:

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

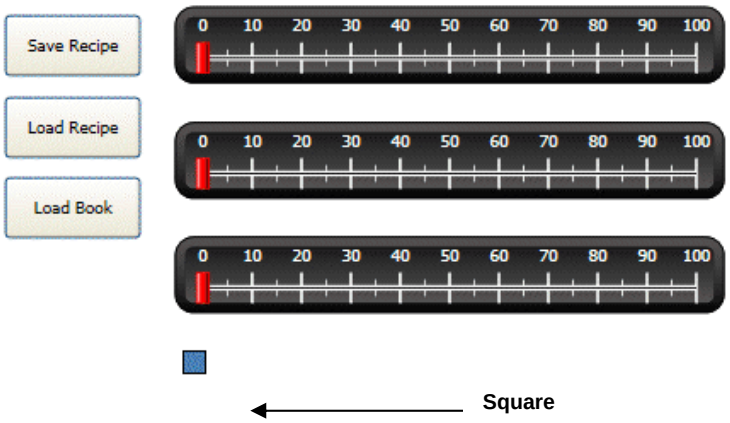
Dynamic object properties are used to move and resize objects based on controller tag values.

This chapter describes how to change the size and color of an object based on tag value changes.

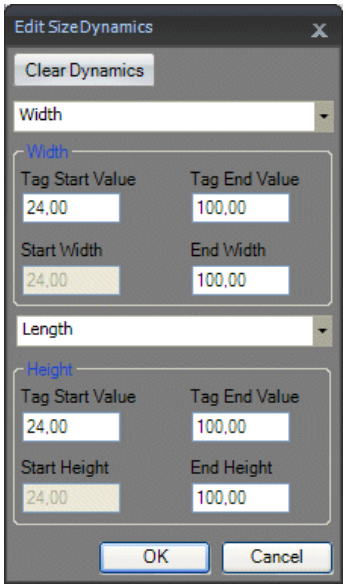
10.1 Resizing an Object

To resize the object that was previously created:

1. Open Screen4 for editing. Select the **Rectangle** in the **Objects** group of the **Home** tab and place a small square below the set of sliders for the recipe tags.



2. Select the square. Click on **Size** on the **Dynamics** tab. Select the **Width** tag for **Width** and the **Length** tag for **Height**.
3. Adjust the enlarged size of the square directly on the screen and note the change of values in the Resize Dynamics Editor window.
4. Set the fields for **Tag Start Value** to the start size values of the square (Start Width, Start Height). Set both fields for **Tag End Value** to **100**.

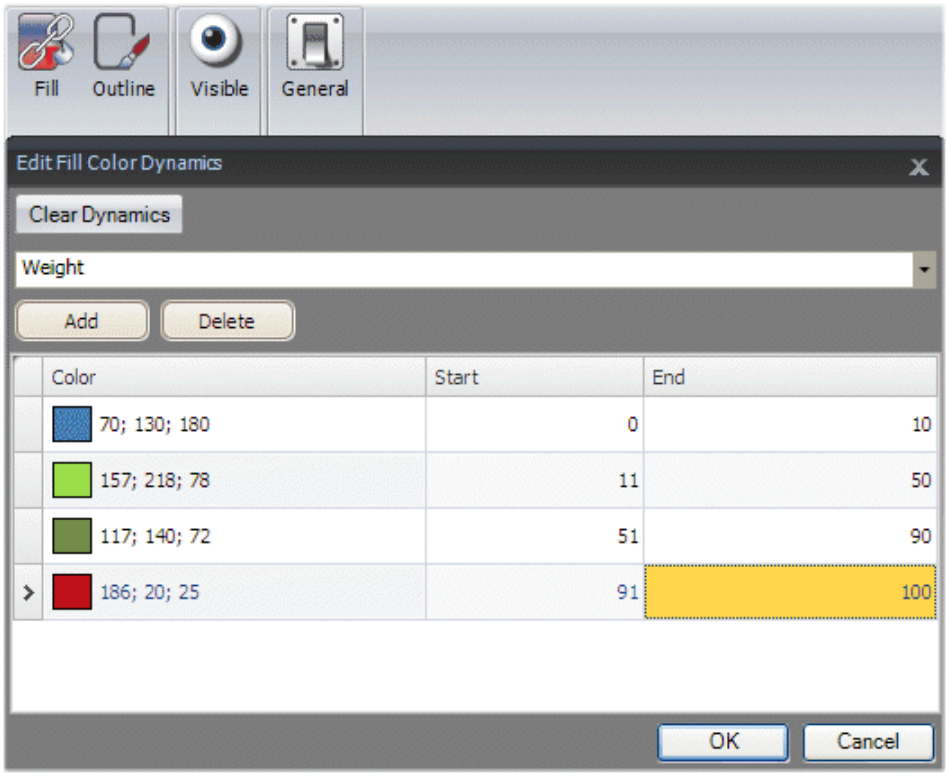


10.2 Coloring an Object

To color an object:

1. Select the square. Click on **Fill** in the **Color** group of the **Dynamics** tab. Assign the Weight tag in the Select Tag drop-down list.
2. 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.



10.3 Running Dynamics Test

To run a dynamic 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 SCRIPTS

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

This chapter describes how to insert a button and write a script for the button to affect the text in the text box.

11.1 Adding Objects

To add an object:

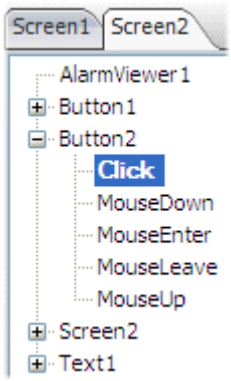
1. Open Screen2 for editing and add a **TextBox** from the **Objects** group (located under Windows Controls) to the screen.
2. Place a button on the screen and label it **Write Test**.

11.2 Script Tab

To create a script:

1. Select the Write Test button.
2. Click on the **Script** tab located in the lower left the desktop area. This will change from Layout view mode to Script view mode.
3. Click on the **Button2** node.
4. Double-click on the **Click** node to 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:




```
...
public partial class Screen2
   {
     void Button2_Click(System.Object sender, System.EventArgs e)
     {
        TextBox1.Text= "Test";
     }
   }
...
```

11.3 Running a Script

To run the script:

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

This chapter describes how to write a script to perform a calculation of the area using the length and width tags and show the result with an internal tag.

12.1 Adding Internal Tags

To add an internal tag:

1. Click on **Tags** in the **Project Explorer**.

   The tags configuration page opens on the desktop.

2. Add a tag and label it Area. Change the data type to **FLOAT**.
3. Add a tag and label it **Calc** and select the data type **BIT**.

	Function Key	Action
	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 ···

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

4. Switch to the Script view mode and locate the Calc tag. Click on the Calc tag node and double-click to open the Value Change node.

   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).

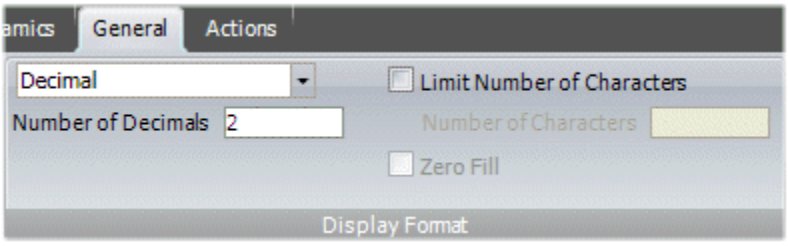
5. Calculate the area in the Value Change node.

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

12.2 Creating an Analog Numeric

To create an analog numeric:

1. Open Screen4 for editing. Place an **Analog Numeric** object below the set of sliders for the recipe tags, clear from the rectangle object.
2. Right-click on the analog numeric and connect it to the Area tag.
3. Keep the object selected, select **Decimal** for **Display Format** and set **Number of Decimals** to **2** on the **General** ribbon tab.



4. Use a **Text** object to place an explaining text (e.g. Area:) in connection with the analog numeric object.

12.2.1 Calculation Trigger

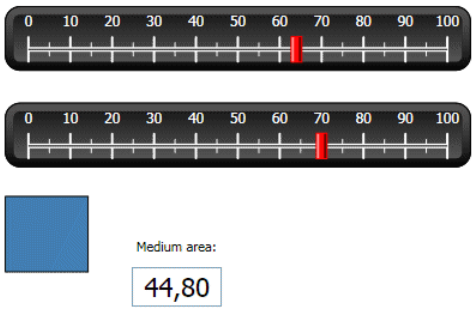
To create an calculation trigger:

1. Open Screen4 for editing. Select the Rectangle object.
2. Go to the **Actions** tab and select **Toggle Tag** from the **MouseDown** drop-down list. Select the **Calc** tag.

12.3 Running Internal Tags Test

To run an internal tag 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.

This chapter describes how to position objects in depth, lock objects, and make objects invisible.

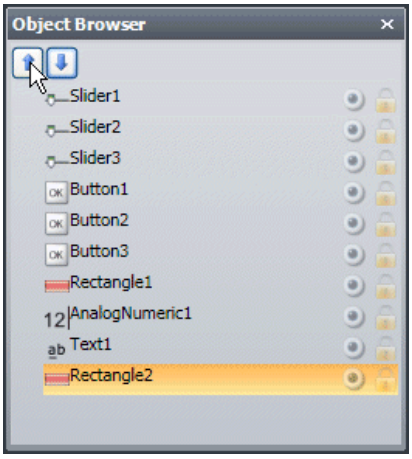
13.1 Adding a Graphical Element

To add a graphical element:

1. Open Screen4 for editing.
2. Select the **Rectangle** from the Objects group of the Home ribbon tab.
3. Resize the 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



To lock and make objects invisible:

1. Select **Object Browser** from the View ribbon tab.
2. Click the **Send object backward** button while the rectangle is selected until all the buttons and sliders are visible.
3. Click the **Lock** button while the rectangle is selected.
4. Try to move the rectangle on the screen by dragging it.

   Note: The object is locked and cannot be moved.
5. Select one of the buttons and click the **Visibility** button.

   The button is hidden. This makes it easy to reuse a screen in different projects with different functionality.

14 MULTIPLE TEXTS

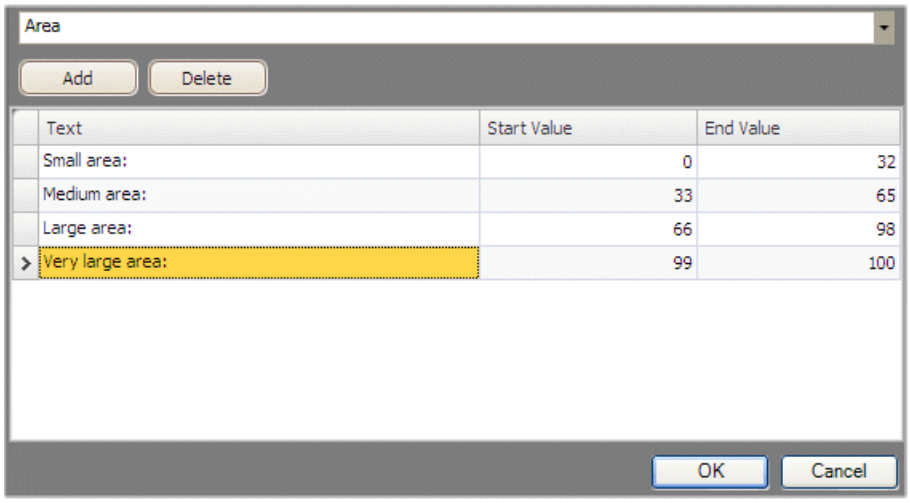
Text objects can be used to display textual information, and can also reflect changes in controller tags.

This chapter describes how to present a variant text message that reflects the changes of the calculated area.

14.1 Configuring Texts

To configure texts:

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

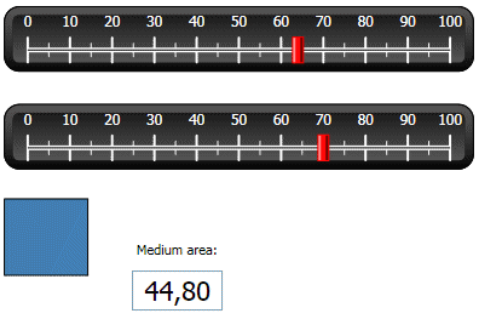


With the default setting for the text object, Autosize, there is no need to adjust the object in order to make the longest string fit in runtime.

14.2 Running Multiple Texts Test

To run 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. This chapter describes how to:

- Add user names and set passwords
- Set up login and logout control
- Restrict access of recipe handling

15.1 Security Configuration

To access the security configuration page:

1. 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

To apply security to users:

1. Select the **Users** tab on the Security configuration page.
2. Add a user that belongs to both security groups; Administrators and Operators.
3. Add another user that belongs to Operator group only.

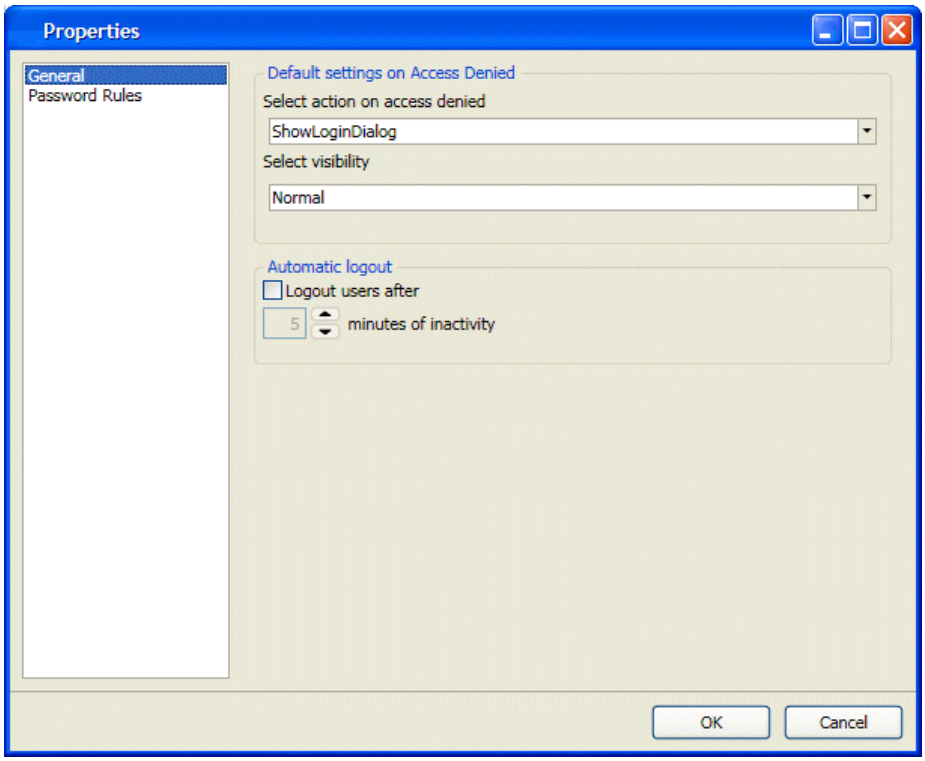
	Username	Password	Description	Groups
	Administrator	*********		Administrators
	SuperUser	*********		Administrators, Operators
I	User1	opxy		Operators

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 **ShowLoginDialog** for action on access denied, and **Normal** for visibility.



15.3 Creating a Logout Button

To create 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. Select **Logout** in the **Click** drop-down list on the **Actions** tab.

15.4 Object Security for Administrators and Users

To set object security for administrator and users:

1. Open Screen4 for editing.
2. Right-click on the **Save Recipe** button and select **Administrators** for **Select Security Groups**.
3. Right-click on the **Load Recipe** button and select **Operators** for **Select Security Groups**.

15.5 Running a Security Test

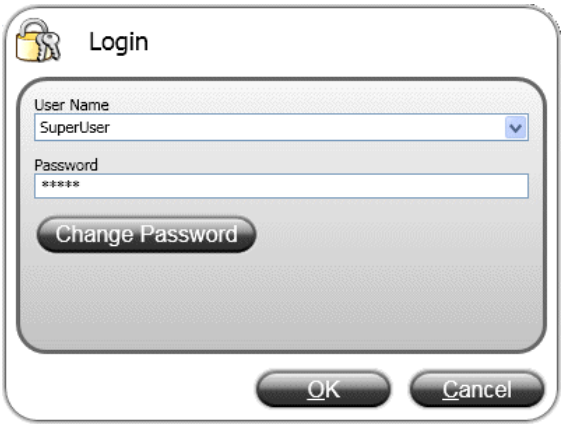
To run a security test:

1. Run the project.
2. 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.

1. Login as **Administrator** and save a recipe.
2. Test to load a recipe.

   The login dialog opens.
3. Login as **User1** and load a recipe.
4. Test to save a recipe.

   The login dialog opens.
5. Login as **SuperUser**. Test to save and load recipes.
6. Log out.
7. 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.

In this chapter you will learn how to:

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

16.1 Defining Function Key Actions

To define a function key action:

1. Click on **FunctionKeys** in the **Project Explorer** to open the configuration page.
2. Click on the row for function key **F1**. Click **Add** and select **Show Start Screen** from the **New Action** dialog.
3. Click on the row for function key **F2**. Select **Show Screen** as action, and **Screen2** from the screen drop-down list.
4. Set up function key **F3** to show Screen4.
5. Click on the row for function key **F4**. Select **Login** as action.
6. Click on the row for function key **F5**. Select **Set Analog** as action, and the **Weight** tag from the **Select Tag** drop-down list. Specify the analog value **50**.
7. Set up function keys **F6** and **F7** to control the **Length** and **Width** tags.
8. Set up function key **F8** to load Recipe1, and function key **F9** to save Recipe1.

   Leave **Recipe data** empty.
9. Set up function keys **F10** to set time zone, region and daylight saving.

Function Key	Action
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**. Select **Script** view mode by clicking on the **Script** tab at the bottom of the screen.
2. Click on the **F11** node, and double-click on its **Click** node.
3. Calculate the area on the click 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 with an **Acknowledge All** for alarms:

1. Select the **Click** node for function key **F12**.
2. Type the following click event code:

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

3. Type the following C# code in the script tab:

```
...
public partial class FunctionKeys
{
   void F11_Click(System.Object sender, System.EventArgs e)
   {
      Globals.Tags.Area.Value =
         (double)Globals.Tags.Length.Value *
         (double)Globals.Tags.Width.Value / 100;
   }
   void F12_Click(System.Object sender, System.EventArgs e)
   {
      Globals.AlarmServer.Acknowledge();
     }
}
```

16.3 Running Function Key Test

To run 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.

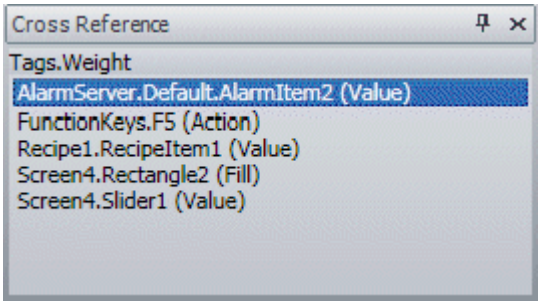
In this chapter you will learn how to locate all occurrences of a specific tag quickly.

17.1 Using the Cross Reference Tool

To use the cross reference tool:

1. Select the **Weight** tag in the Tags configuration page, and click the **Cross Reference** button.

   The Cross Reference tool is displayed.



2. Double-click on the first item in the list.

   The Alarm Server configuration page opens on the desktop.

3. Double-click on the last item in the list.

   Screen4 opens on the desktop, and **Slider1** is selected.

About Kollmorgen

Kollmorgen is a leading provider of motion systems and components for machine builders. Through world-class knowledge in motion, industry-leading quality and deep expertise in linking and integrating standard and custom products, Kollmorgen delivers breakthrough solutions that are unmatched in performance, reliability and ease-of-use, giving machine builders an irrefutable marketplace advantage.

For assistance with your application needs, contact us at: 540-633-3545, contactus@kollmorgen.com or visit www.kollmorgen.com

North America

Kollmorgen

203A West Rock Road
Radford, VA 24141 USA
Phone: 1-540-633-3545
Fax: 1-540-639-4162

Europe

Kollmorgen

Wacholderstraße 40 – 42
40489 Düsseldorf Germany
Phone: + 49 (0) 2039979235
Fax: + 49 (0) 20399793314