
For Classroom Use Only!
Important User Information

This documentation, whether, illustrative, printed, “online” or electronic (hereinafter “Documentation”) is intended for use only as a learning aid when using Rockwell Automation approved demonstration hardware, software and firmware. The Documentation should only be used as a learning tool by qualified professionals.

The variety of uses for the hardware, software and firmware (hereinafter “Products”) described in this Documentation, mandates that those responsible for the application and use of those Products must satisfy themselves that all necessary steps have been taken to ensure that each application and actual use meets all performance and safety requirements, including any applicable laws, regulations, codes and standards in addition to any applicable technical documents.

In no event will Rockwell Automation, Inc., or any of its affiliate or subsidiary companies (hereinafter “Rockwell Automation”) be responsible or liable for any indirect or consequential damages resulting from the use or application of the Products described in this Documentation.  Rockwell Automation does not assume responsibility or liability for damages of any kind based on the alleged use of, or reliance on, this Documentation.

No patent liability is assumed by Rockwell Automation with respect to use of information, circuits, equipment, or software described in the Documentation.

Except as specifically agreed in writing as part of a maintenance or support contract, equipment users are responsible for:

- properly using, calibrating, operating, monitoring and maintaining all Products consistent with all Rockwell Automation or third-party provided instructions, warnings, recommendations and documentation;
- ensuring that only properly trained personnel use, operate and maintain the Products at all times;
- staying informed of all Product updates and alerts and implementing all updates and fixes; and
- all other factors affecting the Products that are outside of the direct control of Rockwell Automation.

Reproduction of the contents of the Documentation, in whole or in part, without written permission of Rockwell Automation is prohibited.

Throughout this manual we use the following notes to make you aware of safety considerations:

---

**WARNING**

Identifies information about practices or circumstances that can cause an explosion in a hazardous environment, which may lead to personal injury or death, property damage, or economic loss.

---

**IMPORTANT**

Identifies information that is critical for successful application and understanding of the product.

---

**ATTENTION**

Identifies information about practices or circumstances that can lead to personal injury or death, property damage, or economic loss. Attentions help you:

- identify a hazard
- avoid a hazard
- recognize the consequence

---

**SHOCK HAZARD**

Labels may be located on or inside the drive to alert people that dangerous voltage may be present.

---

**BURN HAZARD**

Labels may be located on or inside the drive to alert people that surfaces may be dangerous temperatures.
Contents

Before you begin .................................................................................................................. 5
About this lab ......................................................................................................................... 5
What will you accomplish in this lab .................................................................................... 5
Who should complete this lab ............................................................................................. 5
Tools & prerequisites ............................................................................................................ 6
Document conventions .......................................................................................................... 6
FactoryTalk® View Machine Edition .................................................................................... 6
PanelView™ Plus 6 ................................................................................................................ 7
Creating a new Application – Reviewing the basics .............................................................. 7
Launching FactoryTalk View Studio for Machine Edition .................................................. 8
Working with Displays ......................................................................................................... 17
Exploring Display Components ............................................................................................ 24
Customizing the Shutdown button ....................................................................................... 29
Display Navigation ................................................................................................................ 31
Reduce design time by using Global Objects ......................................................................... 35
Using Global Objects on the Runtime Displays ................................................................... 40
Creating a Data Log Model and Trend object to collect and display data ............................ 44
Creating the Data Log Model ............................................................................................... 44
Configure application to start the Data Log Model on startup ........................................... 53
Create the “Trending” graphic display .................................................................................. 55
Add a collection of Trend navigation buttons from the library .............................................. 56
Examine and configure the Trend object properties ............................................................. 60
Modify the application to allow navigation to the “Trending” display ................................. 69
Testing the Datalogging and Trending .................................................................................. 77
Configuring Alarm Triggers and Messages ........................................................................... 79
Examine Advanced Alarm Settings ..................................................................................... 87
Importing and Configuring the Alarming screen ................................................................. 89
Modify the application to allow navigation to the “Alarming” display .................................. 94
Before you begin

During this lab, you can choose to learn about the FactoryTalk® View Studio for Machine Edition (ME) skills that are needed to create a new FactoryTalk® View Machine Edition application. You will review some of the basic and more advanced capabilities and new functionality in FactoryTalk® View ME v6 and PanelView™ Plus terminals.

All sections take advantage of the new PanelView™ Plus 6 terminals that are located at each workstation.

About this lab

This lab will teach you how to design and implement an operator interface for the blending section of a beverage plant using FactoryTalk® View Studio - Machine Edition. You will develop an overview display for the Plant, create global objects for display Navigation, create a Blending display to explore different methods to display and enter data. You will also create a trending display to monitor the tank and ingredient levels with data logging, a recipe to view beverage ingredients, and configure alarms and Information messages to alert the operator when Tank levels are high. Finally, you will put it all together by using Global Object parameters to show information of the blending tanks using one simple faceplate.

This lab takes approximately 90 minutes to complete.

What will you accomplish in this lab

The first part of this lab is a short review of essential components required to create a FactoryTalk® View Machine Edition application. You will:

- Review the basics; create a new Machine Edition application, configure communications, update the default MAIN display and configure a connection to a SoftLogix controller from a numeric display to test the communications.
- Create new displays and explore the use of images and display navigation using global objects to simplify and speed up design time.
- Create a data log model to track the beverage tank levels and current ingredients
- Create an historical trend using the data log model
- Configure Alarms that alert the operator when the tank levels get too high
- Configure Information messages that alert the operator when the tank is 75% full and should be discharged.
- Import an existing recipe for an existing beverage line and add a new recipe for a new beverage line
- Create a faceplate to view the status of 3 blending tanks using Global Object Parameters

Who should complete this lab

Individuals who have completed FactoryTalk View Machine Edition and PanelView Plus Introductory Lab or have had a similar introduction to the basic functionality of FactoryTalk View Machine Edition.
Tools & prerequisites

To complete this lab you must use;

- A Microsoft Windows 7 professional computer
- Ethernet connection between computer and PanelView Plus 6 terminal
- FactoryTalk® View Machine Edition Studio v6.10 (CPR9 SR4)
- FactoryTalk® VuePoint ME v2.10.17
- FactoryTalk® Services Platform 2.50 (CPR9 SR3)
- RSLogix Enterprise v5.40 (CPR9 SR4)
- RSLogix Classic v2.59 (CPR9 SR5)
- RSLogix 5000 v19.01
- SoftLogix 5800 v19.01

Document conventions

Throughout this workbook, we have used the following conventions to help guide you through the lab materials.

<table>
<thead>
<tr>
<th>This style or symbol:</th>
<th>Indicates:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Words shown in bold italics</td>
<td>Any item or button that you must select, click on, or a menu name from</td>
</tr>
<tr>
<td>(e.g., RSLogix 5000 or OK)</td>
<td>which you must choose an option or command. This will be an actual</td>
</tr>
<tr>
<td></td>
<td>name of an item that you see on your screen or in an example.</td>
</tr>
<tr>
<td>Words shown in bold</td>
<td>This is the name of an item that you see on your screen or in an example.</td>
</tr>
<tr>
<td>(e.g., Communication Setup)</td>
<td></td>
</tr>
<tr>
<td>Words shown underlined and enclosed in</td>
<td>An entry that you must type in the specified field. This is information that</td>
</tr>
<tr>
<td>single quotes</td>
<td>you must supply based on your application (e.g., a variable).</td>
</tr>
<tr>
<td>(e.g., ‘Controller1’)</td>
<td><strong>Note:</strong> When you type the text in the field, remember that you do not</td>
</tr>
<tr>
<td></td>
<td>need to type the quotes; simply type the words that are contained within</td>
</tr>
<tr>
<td></td>
<td>them (e.g., Controller1).</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>This is sample text.</td>
<td>Text that appears inside of a gray box is supplemental information</td>
</tr>
<tr>
<td></td>
<td>regarding the lab materials or learning goals; the information is not</td>
</tr>
<tr>
<td></td>
<td>required for you to complete the lab exercises. The supplemental text</td>
</tr>
<tr>
<td></td>
<td>may provide you with helpful hints that can make it easier for you to use</td>
</tr>
<tr>
<td></td>
<td>this product.</td>
</tr>
</tbody>
</table>

**Note:** If the mouse button is not specified in the text, you should click on the left mouse button.

FactoryTalk® View Machine Edition

FactoryTalk® View Machine Edition (ME) is a machine-level HMI product that supports both open and dedicated operator interface solutions for monitoring and controlling individual machines or small processes. It provides a consistent operator interface across multiple platforms, including Microsoft® Windows® CE and 32-bit Microsoft® Windows® 7, XP, Vista solutions. FactoryTalk® View Machine Edition contains two components:
### FactoryTalk View® Studio

This is the development environment containing the tools you need for creating all aspects of a human-machine interface (HMI), including graphic displays, trends, alarm reporting and real-time animation. It also provides tools for testing individual displays and entire applications. When development is completed, a run-time (.MER) file created to run on a PanelView Plus or personal computer.

### FactoryTalk View® Machine Edition Station

This is the run-time environment. FactoryTalk View Machine Edition Station executes the run-time (.MER file) application. FactoryTalk View Machine Edition Station is embedded in PanelView Plus terminals. Run-time applications may also be executed on a personal computer. Executing run-time applications on a personal computer requires additional software licenses.

---

### PanelView™ Plus 6

The PanelView Plus are operator interface terminals designed to optimize system development, performance, and efficiency. The PanelView Plus 6 line is the latest addition to Rockwell Automation’s versatile family of Allen-Bradley PanelView operator interface displays for machine level operator terminal applications in industrial environments.

Enhancements to the hardware platform, embedded operating system and development environment enable users to:

- **Take time and costs out of application development** - PanelView Plus 6 dramatically reduces development, setup time and troubleshooting time through features like tag re-use, complete Symbol Factory graphic library and pre-built face plates.

- **Run their processes more effectively** - Improved hardware performance delivers up to 30% faster screen response, enabling operators to navigate through screens more quickly, and can help avoid maintenance calls resulting from mistakenly pressing inputs multiple times. In addition, new capabilities wring more value from your process.

- **Reduce maintenance costs** - New features like on-board pdf capability and remote user access enable context-sensitive help and can avoid on-site visits to get processes back up and running faster when things go wrong.

---

**Creating a new Application**

About this scenario:

You are an engineer who works for a Systems Integrator. Your customer is adding a production line to his beverage facility. The customer has requested a new application to monitor and control the new line. The customer has a set of standard icons he would like you to use to keep a consistent look and feel with the existing applications in the facility.

In this first section you will review how to:

- Launch FactoryTalk® Studio for Machine Edition
- Create a new project, configure project settings, configure communication to a SoftLogix Controller, and add content to the project
- Test communications by using Test Display
Launching FactoryTalk View Studio for Machine Edition

1. From the Start menu, select Programs > Rockwell Software > FactoryTalk View > FactoryTalk View Studio or Programs > FactoryTalk View Studio as shown below.

After the FactoryTalk Studio for Machine Edition program opens you will see something similar to the following:
2. Click the **New** tab to activate the tab.

3. In the **Application name:** field enter ‘Beverage_ME’. Next, click the **Create** button.

Compact Machine Edition applications are applications intended to run on the PanelView Plus Compact terminal.

Compact Machine Edition applications have the following restrictions:

1) a maximum of 25 displays
2) a maximum of 200 alarm messages
3) a maximum screen resolution of 640x480

Only one RSLinx Enterprise shortcut or single KEPServer Enterprise data server can be used to communicate with a single device. Only serial and Ethernet communications; if the application uses RSLinx Enterprise, only Serial-DF1, Serial-DH485 and Ethernet drivers are supported.

Not supported for PanelView Plus 6

Not supported for FactoryTalk ViewPoint
After creating the application, the FactoryTalk® Studio for Machine Edition opens the application:

At this point, if you are unfamiliar with FactoryTalk Studio for Machine Edition or need a quick refresher, please review the information in Appendix A.

By default, new projects in FactoryTalk Studio for Machine Edition are configured for a PanelView Plus 700 terminal. Your customer has specified a PanelView Plus 6 1250 which has a larger display and resolution, so you need to change the project window size for your project.

4. Double-click the Project Settings item located in the System container to open the dialog.

5. Select the PVPlus 1250 (800x600) item from the Project window size setting using the dropdown list.
6. Click OK

When you change **Project Settings** window size, you are provided with an option to scale existing graphics. When you create a new application in ME v6.10, a default display, **MAIN** is created for you. In this case, we only want to scale the graphic as the fonts, border sizes and images are at the desired size.
7. Select *Scale graphic displays* checkbox shown below only, deselect the sub options.

![Graphic Display Scaling](image)

You have made changes to the project window size. These changes will affect all graphic displays. Select the options below that apply and press OK to continue or press Cancel to return to the Project Settings editor.

- [ ] Scale graphic displays.
- [ ] Scale the font sizes used by objects.
- [ ] Scale the border sizes used by objects.
- [ ] Scale the images in the Images folder

**WARNING:**

If you scale the graphic displays you may not be able to return the application to its original state. It is recommended that you create a backup of your application if you need to restore it to its original state.

![OK Button](image)

8. Click **OK** to scale the application.

Next, you will setup communications

It's good practice to configure communications to the controller(s) at the beginning of the application development. Troubleshooting any problems at this stage can save time before the application becomes more complex.

9. Expand *RSLinx Enterprise* to reveal the *Communication Setup* component. Double-click *Communication Setup*

![Communication Setup](image)
10. Select **Create a new configuration** radio button, and click **Finish**.

Let’s review the Communication Setup.

The **Design (Local)** tab is used to configure communications during development and testing in FactoryTalk View Studio.

This is the communication configuration that Test Display and Test Application use.

The **Runtime (Target)** tab is used to configure communications for the final runtime application that is compiled and transferred to the PanelView Plus or ME Station PC.
11. Create a shortcut. Make sure you are on the Design (Local) tab. Go to the Device Shortcuts pane on the left, and click Add.

![Device Shortcuts](image)

12. Type ‘PAC1’ to replace the New_Shortcut text.

![Device Shortcuts](image)

13. Locate the controller that will be used in the application.
   Expand the Ethernet, Ethernet driver.
   Expand the SoftLogix5800 EthernetIP module, expand the Virtual Chassis
   Select the 1789-L60/A, InstantFizz controller in slot 2.

![Controller Selection](image)
14. Click **Apply**.

You've made the following changes to the shortcut 'PAC1':

- Design (Local) path edited
  - Old: 
  - New: 1789-A17/A Virtual Chassis.InstantFizz

Press Yes to apply changes. Press No to discard changes.

15. Click **Yes** in the confirmation dialog that appears after modifying or creating a shortcut.

Your setup should look like the image below:

For this lab, we will use the **Copy from Design to Runtime** button to configure the Runtime communications.
16. Click the **Copy from Design to Runtime** button to copy the configuration.

Click **Yes** in the confirmation dialog box.

![Copy button](image)

**Warning:** The **Copy** button should be used with care. All devices shown in the **Design** tab will be copied to the **Runtime** tab. The only devices required in the **Runtime** configuration are the controller(s) the PanelView Plus is communicating with.

17. Open the **Runtime (Target)** tab to verify the Runtime configuration.

![Runtime tab](image)

Note the status message at the bottom of the **Communication Setup** window. This is a great tool to ensure your shortcuts are created properly!

Make certain you always click **OK** to close the Communication Setup window after creating or making changes. This ensures the information is written to the shortcut file.

18. Click **OK** to save the communication configuration and close the **Communication Setup**
Working with Displays

Now that you have configured Project Settings and Communication Setup, you are ready to add content to the MAIN display.

1. Expand Displays and double-click MAIN to open the display.

The MAIN display will open in the Work Pane of FactoryTalk Studio for Machine Edition.

You are now ready to add objects to the display.

Adding content to a Display

We are using this display to test communications to the SoftLogix controller.

1. Select the Numeric Display from the Objects toolbar.

To create the Numeric display object, left-click while dragging the mouse down and to the right.

When you move the mouse over the empty display frame, you will note that the mouse pointer changes from the standard pointer to an object pointer to reflect the current selecting tool from the Objects toolbar.
When you release the left-mouse button the Numeric Display object matches the size (height and width) of the box you created while dragging the mouse. The default button has a dark blue background with white caption text.

2. Select the **Numeric Display Properties** by double-clicking on the object.

![Numeric Display Properties dialog](image)

All objects in FactoryTalk View Studio have properties. The properties are organized by function on to tabs on a property dialog. At a minimum, all objects provide a General and Common tab.

<table>
<thead>
<tr>
<th>Tab Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>General</td>
<td>Configure the operation and appearance of the object.</td>
</tr>
<tr>
<td>Common</td>
<td>Configure the object’s size (Height and Width) and display position. Display position locates the top-left corner of the object relative to the top-left corner of the display. Display top-left corner is (0,0) coordinate.</td>
</tr>
</tbody>
</table>

Each object provides a set of tabs that vary with the object type. For instance, most objects that display static text (e.g. Text, Shutdown button) have a Label tab.

<table>
<thead>
<tr>
<th>Tab Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Label</td>
<td>Configure an object’s caption font, font size, font color.</td>
</tr>
</tbody>
</table>
3. Select the **Connections** tab. Then select the **Tag** ellipsis button for the **Value** connection.

4. As this is the first time you have browsed to the newly created Shortcut, right-click on the Application **Beverage_ME** and select **Refresh All Folders**.
5. Expand the shortcut PAC1 and expand and select Online.

6. Select TestTag1A from the right hand ‘leaf’ in the Tag Browser. Note the tag is shown in the Selected Tag area at the bottom of the tag browser.

7. Click the OK button to choose the selected tag for the numeric display.
The following defines and highlights the uses and functionality provided in the Tag Browser.

**Folder List**

<table>
<thead>
<tr>
<th>Folder</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HMI Server</td>
<td>Contains tags defined in the HMI Server; referred to as <em>memory tags</em>. The FactoryTalk Machine Edition runtime manages and updates these tags.</td>
</tr>
<tr>
<td>System</td>
<td>Contains default HMI tags for system information, such as, time of day, date, etc.</td>
</tr>
<tr>
<td>&lt;Communication Shortcut&gt; (e.g. L15)</td>
<td>Contains tags that exist in the memory of the device the shortcut represents.</td>
</tr>
<tr>
<td>Diagnostic Items</td>
<td>Predefined diagnostic and troubleshooting tags useful for monitoring communication and controller status.</td>
</tr>
<tr>
<td>Online</td>
<td>Tags contained in a Logix Controller's memory; the tags are directly referenced by the FactoryTalk Machine Edition runtime.</td>
</tr>
</tbody>
</table>

**Tag Area**

<table>
<thead>
<tr>
<th>Name</th>
<th>Access Rights</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cmd_StopOner</td>
<td>ReadWrite</td>
</tr>
<tr>
<td>Cmd_StopProg</td>
<td>ReadWrite</td>
</tr>
<tr>
<td>EnableIn</td>
<td>ReadOnly</td>
</tr>
<tr>
<td>EnableOut</td>
<td>ReadOnly</td>
</tr>
<tr>
<td>Trip_StopAtPoint</td>
<td>ReadWrite</td>
</tr>
<tr>
<td>Set_AccelOn</td>
<td>ReadWrite</td>
</tr>
<tr>
<td>Set_AccelOff</td>
<td>ReadWrite</td>
</tr>
</tbody>
</table>

**Name**

Column containing the tag’s name. Use the **Tag filter** function to quickly find tags that start with the user-entered filter.

**Access Rights**

New! In V18 firmware, Logix Controllers implements Access Rights for tags. This column displays a tag’s Access Rights property. Tags are either *ReadOnly* or *ReadWrite*. 
The SoftLogix tag is now associated with the **Value** connection of the **Numeric Display**.

![Numeric Display Properties dialog](image)

8. Click the **OK** button at the bottom of the **Numeric Display Properties** dialog.

   We are ready to test the communications!

9. Select the **Test Display** icon from the toolbar

   The numeric display should look like this

   ![Test Display](image)

   Remember that **Test Display** uses the communication configuration from the **Design(Local)** tab, and the runtime application on the PanelView Plus uses **Runtime(Target)** communication. If test display provides valid data, and the application running on the PanelView Plus does not get valid data, check that the **Runtime(Target)** communication is configured correctly.

10. Select the **Edit Display** icon from the toolbar to stop the test display.

   We verified communications are valid to the SoftLogix Controller. We don’t need to save **Numeric Display** object.
11. Click on the **Close** button on the **MAIN** display.
You don’t need to save the **Numeric Display**.
Select **No** in the **Save Changes to MAIN** dialog box.

Congratulations! You have successfully configured and verified communications to the SoftLogix controller.
Exploring Display Components

In this section, you’ll take a closer look at optimizing displays with images and Wallpaper. Your customer has supplied you with images and logos they use as a company standard; first, you will add these images to your application.

1. In the Explorer Window expand Graphics > Images. Note the default images.

2. Right click on Images and select Add Component Into Application
3. Browse to C:\Lab Files\View ME - Fundamental\ViewME\Images and select all the images.

4. Select Open to import the images to the Beverage_ME application. A migration notification dialog will provide status of the import.

Next, you will update the MAIN display and create a new blending display.

5. From the Application Explorer, expand the Displays container and select MAIN
Next, you will use one of the images that your customer supplied as a background to the display to visually indicate it is the MAIN or overview display.

6. Select the **Images** icon from the **Objects** toolbar. Starting at the top left of the display, draw a large rectangle that covers the whole display.
7. Select *overview_bg*, the image will be shown in the preview box.

8. Select OK to accept the image.

   When images are used as a background, it's a good idea to make the image into Wallpaper. Note the new image is on top of the Shutdown button that was created for you. Before you turn the image into Wallpaper, you will move the image to the back.

9. Click on the image to make sure it is selected, then from the menu, select *Arrange>Send to Back* from the menu.
10. From the menu, select **Edit > Wallpaper > Convert to Wallpaper.**

Objects that have been converted to wallpaper cannot be selected or edited until you unlock the wallpaper by selecting Edit > Wallpaper > Unlock All Wallpaper.

Converting objects to wallpaper can help speed up animated graphic displays because the wallpaper does not have to be redrawn each time one of the overlapping objects changes. The wallpaper objects cannot be animated.

The background image is now wallpaper, and the **Shutdown** button is in the foreground.

Next, you will work with images inside of objects.
Customizing the Shutdown button

Creating a consistent look and feel to your application helps the operator find what they are looking for easier. You will standardize the Shutdown button by using the bitmap images provided by your customer.

1. Double-click on the **Shutdown** button to open the **Shutdown Button Properties** dialog.
   Change the **Border style**: to **None** and **Back style**: to **Transparent**.

![Shutdown Button Properties dialog](image1.png)

2. Click on the **Label** tab. Click the **ellipsis** button in the **Image**: section and browse to the `nav_exit_confirm` image.

![Image Browser dialog](image2.png)
3. Click OK.

4. Delete the text **Shutdown** from the Caption then click **Apply**.

We are going to change the size of the button manually to ensure it is square and the image will fit.
5. Open the Common tab. Change the Height and Width to 50

![Shutdown Button Properties dialog]

6. Press the OK button to close the Shutdown Button Properties dialog.

Your button should look similar to this:

![Shutdown button image]

7. Let's save this display. Click the Save icon from the Standard Toolbar.

Display Navigation

You will create a new display for the Blending production area of the Beverage Facility. We'll explore a few display navigation methods and the use of Global Objects to increase application productivity.

1. From the Application Explorer, right click the Displays container and select New.

![Application Explorer with Displays container highlighted]

You'll add a background image to the display and convert it to Wallpaper as you did with the Main display.
2. Select the *Images* icon from the toolbar. Starting at the top left of the display, draw a large rectangle that covers the whole display. Select `blender_bg`, the image will be shown in the preview box.

3. Select **OK** to accept the image.

Make sure the image is selected by clicking on the image.

4. From the menu, select **Edit > Wallpaper > Convert to Wallpaper**.

Let's save this display before we add any navigation buttons.
5. Click the **Save** icon. Type ‘Blending’ in the **Save Component name:** dialog and click **OK.**

![Save Component Name](image)

Now that you have two displays open, we’re going to change the **Work Pane** layout to allow you to select open displays easily.

6. From the **View** menu, check **Workbook Mode.**

   ![Workbook Mode](image)

   Workbook Mode tabs - if you select Workbook Mode in the View menu, the right pane displays tabs at the bottom for each graphic display and/or editor currently open in the Explorer window.

   Note the tabs at the bottom of the **Work Pane.**

   Simply click on the tabs to edit the display.

Now that you have two displays, let’s add navigation buttons.

**Display Navigation:**

**Goto Display**

At run time the operator can press a goto display button to open the graphic display assigned to the button.

**Tip:** The operator cannot use this button to open Replace displays if display change is currently controlled remotely. But the operator can still open On Top displays (with or without the Cannot Be Replaced option).

**Return To Display**

At run time the operator can press a return to display button to close the graphic display that the button is on and reopen the display that was previously open.

**Close Display**

At run time the operator can press a close display button to close the graphic display that the button is on.

**Display List Selector**

Use the display list selector to show a list of graphic displays that the operator can choose from. The operator can scroll through the list and select a graphic display. The operator can scroll through the list and select displays using key buttons or by using the arrow keys and Enter key on a keypad or an external keyboard.
1. Select the *Goto Display* icon from the toolbar or select *Display Navigation > Goto* from the *Objects* menu.

Create a small box about ½” square for the button.

2. Double-click to open the *Goto Display Button Properties* dialog. Change the *Border style* to *None* and the *Back style* to *Transparent*.

3. Select the *ellipsis* button beside the *Display* in the *Display settings* section and select *MAIN* from the display browser. Click *OK*.

4. Click on the *Label* tab. Click the *ellipsis* button in the *Image* section and browse to the *nav_overview* image. Click, *OK*.

We are going to change the size of the button manually to ensure it is square and the image will fit.
5. Open the **Common** tab. Change the **Height** and **Width** to ‘50’.

![Goto Display Button Properties](image)

6. Press the **OK** button to close the **Goto Display Button Properties** dialog.

![Button Image]

Your button should look similar to this

---

**Reduce design time by using Global Objects**

Display navigation is needed on most of the full size (Replace) displays. FactoryTalk View global objects allow you to link the appearance and behavior of a graphic object to multiple instances of that object. When you update the original object, the changes are automatically applied to all the copies of the object. The copies of the base object are called reference objects.

1. Right-click on **Global Objects** in the **Graphics** folder in the Explorer Window and select **New**.

![Global Objects Menu]

2. Select the **Blending** display from the **Work book** tabs.

![Blending Tab]

---

Page 35 of 155
3. Right-click on the **Goto Display** button you configured in the previous section, and select **Cut**.

4. Go back to the **Untitled Global Object display** and right click in the display and select **Paste**.

5. Use the **Duplicate** icon from the **Graphics Toolbar**. A second instance of the button will appear on top of the first button. Move the button so that they are aligned beside each other.

6. Double-click on the second button to open up the **Goto Display Button Properties**. On the **General** tab, click on the **Display:** ellipsis button and browse to the **Blending** display.

7. Click on the **Label** tab. Click the ellipsis button in the **Image:** section and browse to the **nav_blending** image. Click **OK** to close the image browser. Click **OK** to save the new settings.

You now have two **Goto display** buttons that will be used on the replace displays you will create in the next sections. To take advantage of **Global objects**, you will create a group so that the global object properties can be configured once for all objects in the group.

8. With the **Select** tool, draw a rectangle around all the navigation buttons on the **Global Object Display**.
Notice there are handles around each object.

9. Select the **Group** icon from the **Graphics Toolbar**

Note there is only one group of handles indicating the objects are grouped.

10. Right-click on the new grouped object and select **Property Panel**
Notice the GroupName is **Group1**. It is good practice to give groups used in **Global Objects** a logical name as this name is used when linking to each instance on graphics displays. If the name is not the same, the link will be broken.
11. Change the **GroupName** to ‘NavGroup’

![Property Panel](image)

12. Close the **Property Panel** by clicking the small x in the top right.

You will see the following warning message:

![FactoryTalk View Studio Warning](image)

FactoryTalk View will warn you when you change a property that may affect any linked objects.

In this case, we haven’t used the Global Object anywhere, so there aren’t any reference objects.

13. Select **Yes**. If needed, click the x again to close the **Property Panel**.

Next you will save the **Global Object display** before you use the objects in our displays.
14. Select the **Save** icon from the menu. Enter ‘Navigation.GO’ in the **Save** dialog box, then click **OK** to close the dialog.

**Using Global Objects on the Runtime Displays**

You will copy the navigation buttons base group objects and create a reference object on the **MAIN** and **Blending** displays.

If you are on another display, click on the **Navigation.GO** global object display tab in the **Work pane**

1. Select the **Goto buttons**; right-click and select **Copy** or select **Ctrl + C**.

2. Click on the **MAIN** display tab (or if it is not open, double-click the **MAIN** display in the project explorer), right-click in the display and select **Paste** or press **Ctrl + V**

   A reference of the global objects will be placed on the display.

   Remember you made the **overview_bg** image in to **Wallpaper**, so you don't have to worry the image could be selected when editing!

3. Save the **Main** display by clicking the **Save** icon from the Standard toolbar.
4. Click on the **Blending** tab and then right-click in the display and select **Paste**.

![Blending Display](image)

5. Save the **Blending** display by selecting the **Save** icon from the standard toolbar.

Now that you have created a few displays with navigation buttons, we’ll check on the application’s **Startup** configuration.

6. To open the **Startup** dialog, double-click on the **Startup** item in the **Explorer Window**.

   New in FactoryTalk View Studio v6.0; the **MAIN** display is automatically configured as the **Initial graphic** when the application is created.
7. Press **Cancel** to close the **Startup** settings and accept the default **Initial graphic**.

![Startup settings dialog](image)

The **Startup** dialog is where you can configure various options that will occur when a Machine Edition application first starts. A grey checkbox and description indicate that a component has not been created. A blank checkbox indicates there are components available, but none have been selected.

8. Click the **Test Application** icon from the Standard toolbar.

![Creating runtime file dialog](image)

When you test an application, any opened and saved graphic will close. Any display that has unsaved changes will prompt you to save the display.

A **Creating runtime file** dialog box will notify you of the status. This may take a few minutes.
When the runtime file has been created, you should see something similar to the following

Test the navigation buttons we created.

9. Select the **Blending Goto Display** button.

10. From the **Blending display**, select the **Main Goto Display** button

11. When you are done navigating, select the **Shutdown** button

Congratulations! You have created two displays and a Global Object display to provide navigation to your application. Now you are ready to add trending to allow the operator view the status of the blending tanks
Creating a Data Log Model and Trend object to collect and display data

Your customer requested a way to monitor real time tank levels, as well as have an historical record of the same data.

In this section you will learn how to

- Create a Data Log Model and understand the principles of data logging
- Create a Trend object and understand the basic principles of trending

Creating the Data Log Model

1. Right-click the Data Log Models option located in the Data Log System container to open the dialog and select New.
After selecting **New** the following dialog box will appear:

- Data Log Models allow for historical trending.
- Multiple Data Log Models can be defined at design time.
- Only 1 Data Log Model can be enabled at runtime.
- The model will not log data for string tags, array tags, parameters or expressions, and you cannot use tag placeholders in your data log model.
- The Data Log file is saved in a format designed for runtime performance and file size reduction. The file cannot be saved/exported to CSV or other file format at runtime.
2. On the **Setup** tab, type ‘**Beverage Blending**’ in the **Description** field. Enter ‘3000’ in the **Maximum data points**: field.

![Screenshot of the setup tab with Description field set to 'Beverage Blending' and Maximum data points set to '3000'.]

Note that the **Description** field is for your reference only and is not used by Machine Edition.

The **Maximum data points** are the total number of tag values to store. When this number is reached, the oldest tag values are deleted to make room for new values. The minimum is 100; the maximum is 300,000. The default is 1000. The higher you set the limit, the more space the data log file requires on the runtime computer, or, if you are logging to a custom path on a remote computer, on the remote computer. The data log file is created at startup of the application. 300,000 points will use approximately 4.7MB of space.

In this lab you will be logging 5 tags at a rate of 1 second, as a result a **Maximum data points** setting of 3000 will allow us to log approximately 10 minutes of historical data. 3000 data points / 5 data points every second = 600 seconds = 10 minutes.
3. Select the **Paths** tab, here you can define where the Data Log is stored. Leave the path set to the default location.

- If the Path is set to `\Storage Card2\Logs` and the CF or SD card is removed Data Logging will stop.
- Once the data log starts it cannot be stopped during runtime by the operator
- Machine Edition does not support dynamic log file creation during runtime
- Best practices for a Panel View Plus 400-600 terminals is to data log to a external CF card or remotely
- If using a network location the network must have domain name resolution, such as a DNS server, to resolve the UNC computer name defined in your path, you cannot use the IP address of a network location.
4. Select the **Log Triggers** tab; here you can define the method used to trigger the logging of data.

![Log Triggers Tab](image)

**Periodic:**
The fastest log rate possible is 100 ms; however, logging too fast can cause performance related issues. In most cases this rate is not feasible with a typical application due to other loads on communications.

The best practice is to log at the slowest possible rate desired by the process being monitored.

5. Select the **On Change** radio button and examine the options available.

![On Change Tab](image)

**On Change:**
Use the On Change trigger to log tag values for any tag whose value has changed by a certain percentage. A Change Percentage of 0 will log all changes for a tag.

The Maximum update rate is used to specify the maximum rate at which data servers will send data to the tags in the data log model.

The Change percentage is the percentage a tag value has to change in order to trigger logging. To log all changes, enter 0.

**Heartbeat**
Type a time and select a time unit to specify how often tag values are logged even if no change has occurred. The heartbeat is ensures that the data in the log file is current.
6. Re-select the **Periodic** radio button, enter *1* in the *Interval* field as shown below, and leave the default at *Seconds*.

![Screenshot of Data Log Model setup with Periodic and Interval fields highlighted.]

7. Select the **Tags in Model** tab; here you will define which tags will be included in the **Data Log Model**.

8. Click the **Tag browse** button to open the tag browser.

![Screenshot of Tag browser window with Tags to add field and Tags in model tab highlighted.]

- Maximum number of tags allowed in a model is 100
- The model will not log data for string tags, array tags, parameters or expressions, and you cannot use tag placeholders in your data log model.
- Tags can be HMI or Direct Reference
9. Expand the **PAC1** shortcut, expand the **Online** folder and navigate down to **Program:Blending.Recipe**. Use the mouse and **Ctrl** key to select the **Syrup** and **Water** tags.

![Tag Browser](image)

10. Click **Add Tags(s) to List**. The 2 tags will now appear in the **Selected tag(s)** window as shown below.

![Selected Tags](image)
11. Within the Tag Browser continue selecting tags by navigating to Program:Blending.Tank1 and select Level.

12. Click Add Tags(s) to List. The Program:Blending.Tank1.Level tag will be added to the list.


At this time you will have 5 tags selected as shown below.

14. Click OK to close the tag browser.
At this point all 5 tags have been selected but have not yet been added to the Data Log Model.

15. Click **Add** to add the 5 tags to the **Data Log Model**.

![Add button image](image1)

You should have 5 tags in the Data Log Model as shown below.

![Tags in model image](image2)

16. Click **Close** to save the **Data Log Model** changes.

![Close button image](image3)
17. Click **Yes** when prompted to save changes.

![FactoyTalk View Studio dialog window](image)

18. Enter ‘Blending Data Log’ in the **Component name** field when prompted.

![Save dialog window](image)

19. Click **OK** to save the Data Log Model.

**Configure application to start the Data Log Model on startup**

1. Double-click the **Startup** option located in the System container to open the dialog.

![System container with Startup option highlighted](image)

After selecting **Startup** the **Startup** dialog box will appear as shown below.
2. Check the **Data Logging** checkbox.

3. Select the **Data Logging** dropdown list button next to the **Data Logging** field and select the **Blending Data Log** you created earlier. Ensure that **MAIN** is selected as the **Initial Graphic**, use the dropdown list button and select **MAIN** if it is not already selected.

4. Click **OK** to save the **Startup** changes.
Create the “Trending” graphic display

You are now ready to create a trend to display the real-time and historical data you configured in the Data Log Model.

1. Right-click the Displays option located in the Graphics System container to open the dialog and select New.

A new Graphic Display will open after selecting New.

2. From the Objects > Trending menu option select Trend.
3. Use the mouse to select a location in the upper left corner of the new display and drag the mouse across to somewhere on the lower right side of the display, be sure to leave some room at the bottom for adding Trend control buttons.

Your new Trend object should look similar to the one shown below:

Add a collection of Trend navigation buttons from the library

1. Expand the **Libraries** option located in the **Graphics** System container.
2. Scroll down and double-click *Trends*

3. Use the mouse to drag a selection window around all the *Trend Controls* at the bottom of the display.
4. From the **Edit** menu option select the **Copy** option to ensure that all objects are copied together.

![Copy Option](image1.jpg)

5. Return to the display containing the new **Trend** object, right-click the display and select **Paste**.

6. Use the mouse to move the group of pasted buttons to the bottom of the display as shown below.

![Trend Display](image2.jpg)

At this point, you will save the changes made so far to the **Trending** display.
7. Click the **Save** button from the **Standard Toolbar** in the upper left corner of Studio.

8. Enter ‘Trending’ in the **Component name** field and click **OK**.
Examine and configure the Trend object properties

1. Double-click the Trend object and the Trend Object Properties dialog will open.
2. Select the **Connections** tab.

Here you will assign the same 5 tags that you selected for the Data Log Model in the previous lab section.

3. Click the tag browser button to the right of **Pen 1** to open the tag browser for **Pen 1**.
4. Expand the **PAC1** shortcut, expand the **Online** folder and navigate down to **Program:Blending.Recipe**. Select the **Syrup** tag and click **OK**.

5. Perform the procedure used in step 4 above to select the following pen/tag combinations:
   - **Pen 2 = Program:Blending.Recipe.Water**
   - **Pen 3 = Program:Blending.Tank1.Level**
   - **Pen 4 = Program:Blending.Tank2.Level**
   - **Pen 5 = Program:Blending.Tank3.Level**
Once you have selected tags for all 5 pens your Connection tab should look like the one shown below.

Note that only 8 Connections/Pens can be configured in the Trend object.
6. Select the **Pens** tab.

From the **Data Log Model** drop down list, select **Blending Data Log**.

- The Trend object doesn’t support dynamically making pens visible or invisible.
- Note that each pen has a default color; these can be changed.
- The default width of 1 should not be changed as performance issues can result.
- Only one Data Log Model can be loaded and available at runtime.
- The data log will read the data log model only once when opening a display with a trend object.
- 300,000 point data logs can take several seconds to load data and allow the operator to interact with the Trend object. This will occur each time a display is opened that is accessing the Data Log.
- Having several Trend objects on the same display all using the same Data Log will cause performance issues when the display opens.

Note that if no Data Log is selected you will only see real time data in the Trend object. The data will start trending when you open the graphic display and will restart every time you open the graphic display containing the Trend object.
7. Select the **X-Axis** tab.

Notice the default value of 2 in the **Time Span** field. Remember that the **Data Log Model** has been configured to log approximately 10 minutes of data so you will leave the default at 2 minutes to show some of the Data Logging capabilities within the timeframe of the lab.

Setting the X-Axis time span too high (hours, days) will also leave a severe memory footprint. The trend display should never be left open if the X-Axis is configured for hours or days because the HMI will run out of available memory.
8. Select the **General** tab.

You will use the default **Refresh Rate** of 1 second.

Best practice is to set the Trend Refresh Rate no faster than the amount of time it takes to put a data point on a trend

**Formula:** Fastest Possible Refresh Rate = (X-Axis Time (seconds)) / (Width of Trend Object in pixels)

Select the **Y-Axis** tab.
9. Change the **Minimum / maximum value options** to **Custom**.
Leave the radio button on **Actual minimum value** and enter ‘0’ as the minimum value.
Leave the radio button on **Actual maximum value** and enter ‘1000’ as the minimum value.
The **Y-Axis** should look like the example below.
10. Select the **Display** tab.

Highlight the value of **200** in the **Buffer for extra data** field and enter the value **'0'**.

- Note that in this lab we are using a Data Log Model to buffer historical data so we do not need a buffer. If this was a real time trend with no Data Log model and we did not define any buffer you would not be able to scroll back in time and see historical data. Once you attempted to scroll back in time or ahead in time all data on the screen would be lost.

- Using a Data Log Model also allows for data to be seen immediately when the Trend display is opened, depending on how long the application has been running. Without a Data Log Model you will have to wait for real time data to fill the Trend display. If the X-Axis time span is a long time span you will have to wait that long to see how the data is trending.

- Trending memory is consumed over time and not allocated on startup of the application.

- Max buffer size 32767 records.

- Buffering too much extra data will leave a severe memory footprint.

- If the application uses a data log you don’t need to buffer extra data (Data log becomes the buffer).

- The Trend object must be linked to the data log file for Historical Trending.
11. Click **OK** to save changes and close the *Trend Object Properties* dialog.

**Modify the application to allow navigation to the “Trending” display**

The existing Global Objects display navigation bar containing *Goto display buttons* will be edited to allow navigation to the *Trending* display.

1. Open the *Navigation_GO* global object display.

2. Select the navigation button group, then select the *Ungroup* icon from the *Graphics* toolbar.

   ![Ungroup icon](image)

   While you can edit a grouped object, it is sometimes more efficient to ungroup the object, modify it, then group all the objects again.

   Make certain your object has the same *GroupName* property as the original so you don’t break the link between the global object and the copies or instances of the object on graphic displays. You will review this property in an upcoming step.

   Each Goto display button is now a separate object.

   Next, you will add a new button and icon for the *Trending* display.

   3. Click on the *Blending* icon

   ![Blending icon](image)

   4. Use the *Duplicate* icon from the graphics menu to make a new instance of a *Goto display* button.

   ![Duplicate icon](image)
5. Move the new button beside the other buttons.

You may notice that the buttons are not spaced or aligned properly. You will use more display tools to space and align the buttons after we make changes to the new buttons properties.

6. Double-click on the new button to open its properties,

7. Select the **Display** browse button and select **Trending** and press **OK** to accept the new display.
8. Select the **Label** tab and click the browse button to change the **Image**

In the Image Browser press **Add from File...**

You will browse to a directory where images have been placed for you to complete the lab, but first let's take a look at the default directory that opens when you browse for an image:

In Windows 7 and Vista, the browse opens to `C:\Users\Public\Public Documents\RSView Enterprise\Images`.

You can create your own library of images if you re-use them in several applications. Placing them in this directory will allow you to easily browse for the images installed with FactoryTalk View plus any you add yourself.
9. Browse to \LabFiles\ViewME - Fundamental\ViewME

10. Select the `trending_48.bmp` and click **Open** to add it to the application.

![Image Browser](image1)

Note the size of this image is 48x48. Recall that the Goto Display button is 50x50. You can scale images to fit the object you are using them in.

11. Click **OK** in the **Image Browser**

12. Select the **Image Scaled** checkbox

13. Click **OK** in the **Goto Display Button** Properties
14. Select all of the buttons you want to align.

15. Click **Space Horizontal** from the **Arrange** menu item.

16. Next, Click **Align Middle** from the **Arrange** menu item.
Now, you will group the buttons again.

17. Select the **Group** icon from the **Display** toolbar

![Group icon from Display toolbar]

Next, you will verify the new group will be linked properly with each instance on the displays.

18. Select **Property Pane** from the **View** menu

![Property Pane from View menu]

Note the **GroupName** may be something like **Group1**.

![GroupName in Property Pane]

Recall that the **GroupName** must be the same as what is used to link the reference objects.
19. Change the GroupName to NavGroup

![Property Panel]

20. Click the x icon in the Property Panel to close it.

As previously, if you change a property, you will see a warning message:

![FactoryTalk View Studio]

FactoryTalk View will warn you when you change a property that may affect any linked objects. In this case, we want the GroupName to match what we had before, so select Yes. If needed click the x again to close the Property Panel.
21. Click the **Save** icon from the Standard toolbar to save the **Global object** display.
   Now we will use the saved and modified **Global Object** on the **Trending** display.

22. Select the group and press **Ctrl+C**

23. Select the **Trending** tab to bring it to the foreground.
24. Press **Ctrl+V** to paste the global object

Your display should look something like this.

![Display Image]

Let's save the display, then take a look at the previous displays that reference the global object.

25. Select the **Save** icon to save the **Trending** display.

**Congratulations!!** You have successfully created a Data Log Model; created a graphic display containing a Trend object and navigation buttons, created display navigation for the Trending display and configured the Data Log Model to run at application startup.

**Testing the Datalogging and Trending**

1. Click on the **Test Application** (running man) icon, select **Yes** if you are prompted to save.

When the **MAIN** display opens, notice the changes you made to the global object in **Navigation_GO** have been updated.
2. Click on the **Trend** icon.

**NOTE:** Below is a sample screen shot, your screens will show different times

When the application has been running for several minutes, try using the **Move Left, Home, End** and other trend buttons.

Use the Navigation buttons to change to other displays, such as **MAIN**, then return to the **Trending** display. Notice the pens values are still shown for the full two minutes and you are able to move left and right to access the historical data.

3. When you are done reviewing the **Trend**, navigate to the **MAIN** display and click the **Shutdown** button to exit the Test application.
Configuring Alarm Triggers and Messages

In this section you will create a bit alarm trigger using the “L” modifier; you will also create a Value trigger.

1. Double-click the **Alarm Setup** option located in the **Alarms System** container to open the dialog.

2. On the **Triggers** tab click the **Add** button to add an alarm trigger.
3. Click the tag browse button.

4. Expand the **PAC1** shortcut, expand the **Online** folder and navigate down to **Program:Blending.AlarmArrayDint**; Select the **AlarmArrayDint[0]** tag and click **OK**.
5. After the `{::PAC1}Program:Blending.AlarmArrayDint[0]` has been selected use the mouse and cursor keys to highlight the field and edit the tag to add ‘,L2’ after the `[0]` as shown below, the tag should read -

`::PAC1}Program:Blending.AlarmArrayDint[0],L2`

### Tips For Using Array Trigger Tags

**What is the L Modifier?**
- The L modifier is an additional parameter added to the Trigger Tag which allows the alarm subsystem to interpret the tag as an array.
- **Example Syntax:** `{tagname,Larraylength} {::[PAC1]Dint[0],L2}
- The L2 modifier will return two arrays (elements) of the tag. In this example, bits 0 – 63 are scanned. This corresponds to Dint[0] (32 bits) and Dint[1] (32 bits).

**What are the benefits of the L Modifier?**
- Allows for simultaneously triggered alarms
- No handshaking required by PLC
- Monitors individual bits as a trigger
- Returns an array
- Reduced trigger tags
- Efficiency of a value trigger

Note you can also select the `Exprn` browse button to open the expression editor for easier addition of the `.L2` modifier.
6. Click OK to save the changes.

7. Select the **Trigger type** drop down list and select **Bit** as the Trigger type.

The two main trigger types used most often are:

- **Value** - integer or floating point values. Floating point values are rounded to the nearest integer. Use with analog or digital tags.
- **Bit** - a bit array consisting of one or more bit positions. Use this trigger type to generate multiple alarm messages with a single tag (or array tag) or expression. Each bit in the array whose value changes from 0 to 1 triggers an alarm (if a message is configured for the bit).

You will now add another alarm trigger to the application.
8. Click the **Add** button to add another alarm trigger.

9. Click the **tag browse** button.

10. Expand the **PAC1** shortcut, expand the **Online** folder and navigate down to

    `Program:Blending.AlarmArrayDint`; select the **AlarmArrayDint[3]** tag and click **OK**.

    Click **OK** again to accept the tag.

![Trigger Window](image)

Leave the Trigger type for `::{PAC1}Program:Blending.AlarmArrayDint[3]` as the default of Value.

![Alarm Setup Window](image)

11. Select the **Messages** tab.
12. Use the mouse to select and drag the **Trigger** column to widen it and view the full trigger names.

13. Click in the **Trigger** column for **Alarm message 1**, then click the drop down list and select \(::{[\text{PAC1}]\text{Program:Blending.AlarmArrayDint[0].L2}}\).

14. Highlight the **Trigger value** field and enter ‘1’.

15. Highlight the **Message** field and enter ‘Dint[0] bit 0 in alarm’.
16. Use the scroll bar to go back to the Trigger column. Click in the Trigger column for Alarm message 2, then click the drop down list and select {::[PAC1]Program:Blending.AlarmArrayDint[0],L2}.

17. Highlight the Trigger value field and enter ‘37’.

Note: This trigger value is the 37th bit in the array – which corresponds to Dint[1].4.

19. Use the scroll bar at the bottom to go back to the Trigger column.
20. Click the drop down list and this time select \{::[PAC1]Program:Blending.AlarmArrayDint[3]\}.

21. Highlight the **Trigger value** field and enter '95'.

22. Highlight the **Message** field and enter 'Tank level has reached 95%'.

![Diagram of Alarm Setup window](image-url)
Examine Advanced Alarm Settings

1. Select the **Advanced** tab; we will make no changes however take time to examine the options available here.

   Notice that the **Maximum update rate (second)** defaults to 1, this means that all alarm trigger tags will be put on scan to update every second. Notice that the **History Size** defaults to 128, this means that only 128 alarms will be retained in the alarm history, the alarm history is also referred to as the alarm log file. Note that [ALARMS] is configured in the **Current alarms** drop down list this is the alarm display that will pop-up when an alarm is triggered.

   ![Alarm Setup Dialog](image)

   The **History Size** can be configured up to 10,000 alarms. When the history contains this number of alarms, the oldest alarms are deleted when new alarms occur.

   Maximum update rate, select the maximum rate at which data servers will supply data to the trigger tags specified in the Triggers tab. This rate also applies to any tags used in trigger expressions, remote tags or expressions, and tags in trigger message embedded variables. The default update rate is 1 second.

   At run time, all of these tags are updated immediately upon startup, no matter what the update rate is.

2. Click **OK** to close the **Alarm Setup** dialog.
Alarming features and Performance Impacts

- Embedded Alarm Messages
- Maximum Update Rate
- Number of Trigger Tags

Why Do Embedded Alarm Messages Impact Performance?

- Embedded variables constantly update in the background during runtime
- Additional overhead required for multi-tasking
- Scan rate related directly to the “Maximum Update Rate”

Why Does the Maximum Update Rate Impact Performance?

- Controls the rate at which the HMI scans Trigger Tags and Embedded Alarm Messages
- The faster the scan rate the less time available to do other tasks

Alarming Memory Impacts

Alarm Triggers

- All alarm triggers are cached into RAM on startup
- Memory consumption could be anywhere from a few hundred Kb or Mb’s
- Memory usage is related to the number of trigger tags on scan
- Excessive alarm triggers may result in longer application load times
- A dialogue “Starting Alarms” will appear until the operation is completed

Alarm Messages

- Max alarm message length 255 characters
- Lengthy alarm messages consume additional memory
- The alarm message file is built into the .mer file
- The larger the .mer file the more internal storage space consumed

Alarm History Size

- Alarm history is capable of consuming ~1 Mb of internal storage space

Alarm Communication Bandwidth

Why Optimize Trigger Tags?

- Scattered trigger tags will require additional communication packets
- Additional communication overhead is consumed
Importing and Configuring the Alarming screen

You want to maintain the same look and feel as the other applications in the facility. In this section, you will import a display.

1. Right-click the Displays option located in the Graphics system container and select Add Component Into Application....

2. Navigate to the C:\Lab Files\View ME - Fundamental\ViewME\Application Files folder.
3. Select **Alarming.gfx** and click **Open**.

The **Alarming** display will be migrated into the application.

4. Expand the **Displays** option located in the **Graphics** System container and double-click the **Alarming** display.

Similar to what you did in the previous lab section on **Trending**, the button bar at the bottom of the **Alarming** display was copied for you from the **Libraries** object called [HISTORY] 800x600.
5. Select the **Objects > Advanced > Alarm > Alarm List** menu option.

Note that there is a selection of objects associated with Alarming, the top section is a list of buttons dedicated to working with Alarms, the bottom three objects are used to display Alarms in different ways.

**Alarm List object** - shows a list of triggered alarms

Use the alarm list to notify the operator when a situation requiring immediate attention occurs. For example, the list could display a message that warns the operator that the pressure in a boiler is too high.

**Alarm Banner object** - shows a single triggered alarm

An alarm banner is a list displaying only one unacknowledged alarm. The Alarm Banner is typically configured as the display that will pop-up when an Alarm is triggered.

**Alarm Status List object** - can show the status of all alarms in the system

Use the alarm status list at run time to see a list of all alarms, currently active alarms, or alarms that have been active since the alarm status was last reset.
6. Use the mouse to drag a window at the top of the **Alarming** display to place the **Alarm List object** as shown below.

![Alarm List object](image)

7. Double-click the **Alarm List** object and examine the properties available under the various tabs. We will use the default settings.

The Alarm List object has various properties that can be used to configure how alarms are displayed at runtime. Some of the more powerful properties are:

**Filtered triggers**

Click the browse button to open the Trigger Label Selector, where you can select the alarm trigger labels whose alarms will be included in the display as they occur.

**Select alarm condition:**

**Active and acknowledged**

Select this list item and check the Display box to display alarms that are both active and acknowledged.

**Active and unacknowledged**

Select this list item and check the Display box to display alarms that are active but not acknowledged.

**Inactive and acknowledged**

Select this list item and check the Display box to display alarms that are no longer active but are acknowledged.

**Inactive and unacknowledged**

Select this list item and check the Display box to display alarms that are no longer active and have not been acknowledged.
9. Click the X in the upper right corner of the Alarming display to close the display.

10. Click Yes when prompted to save changes.
Modify the application to allow navigation to the “Alarming” display

You’ve modified the Navigation_GO Global Object display in the Trending section. To save time, we’ve created a finished Navigation_GO Global Object display. Next you will add this finished Global Objects display into the application.

1. Right-click on Global Objects in the Graphics folder in the Explorer Window and select Add Component Into Application.

Browse to C:\Lab Files\View ME – Fundamental\ViewME and select Navigation_GO.ggfx and select Open. You will be prompted to overwrite the existing global object display

2. Select Yes to overwrite the existing global object display

3. Open the Navigation_GO global object and view the updates.

We’ve added an Alarm and Recipes icon you will use later in the lab.

4. Click the X in the upper right corner of the Navigation_GO display to close the display.

Congratulations!! You have successfully configured Alarms, created an Alarm List object on a display and imported an updated Global Object display that has pre-configured button to open the Alarming display.

Testing Alarming

You can test the Alarming section of the application now on the desktop, or wait until the lab is complete to test all components together on the PanelView Plus.
1. Select **Test Application** icon from the toolbar.

2. On the **Main** display click the **Alarming** GoTo button to navigate to the **Alarming** display.
3. Click on the button. This is a Numeric Input Enable object that writes a value to the \{::[PAC1]Program:Blending.AlarmArrayDint[3]} tag address which you configured as a Value trigger in Alarming.

4. Enter a value of ‘95’, this is the value you entered for the trigger when configuring the Alarm messages in the Alarm Setup dialog.

5. Click the return key to write the value, trigger the alarm and close the pop-up keypad.

The Tank level has reached 95% alarm message will display in both the Alarm List and the pop-up alarm display.
6. Click the Close button to close the pop-up alarm display.

7. Use the **ACK Alarm** and **Clear All** buttons on the **Alarming** display to work with alarms in the **Alarm List** as they are triggered.

![Alarming display with Ack Alarm and Clear All buttons](image)

**NOTE:** If you want to trigger more alarms for the `{::[PAC1]Program:Blending.AlarmArrayDint[3]}` alarm trigger you will need to perform steps 1, 2, 3 above and this time enter the value of ‘0’ to reset this alarm. After the trigger tag has been set back to 0 a new value of ‘95’ can be entered again to trigger another alarm.

8. Click on the button. This is a Momentary Pushbutton object that momentarily writes a value of 1 to the `{::[PAC1]Program:Blending.AlarmArrayDint[0]}` tag address. As this is a DINT data type in the Controller and you are writing a 1, the value is written to the first bit position in the `{::[PAC1]Program:Blending.AlarmArrayDint[0],L2}` alarm trigger you defined in the **Alarm Setup** dialog. As a result the **Dint[0] bit 0 in alarm** message will appear in the **Alarm List** object and the alarm popup.

![Momentary Pushbutton object](image)

Note that you may have to hold the button down until the Alarm system sees the tags value change. The system response time depends on the network speed and the **Maximum update rate** set in the Alarm Setup.

![Tag address](image)

The **Dint[0] bit 0 in alarm** alarm message will display in both the Alarm List and the pop-up alarm display.
9. Click the **Close** button to close the pop-up alarm display.

10. Click on the button. This is a Momentary Pushbutton object that momentarily writes a value of 1 to the `::{PAC1}Program:Blending.AlarmArrayDint[1]/4` tag address. Note that in this case you are specifying a specific bit within a DINT. This bit is in the second DINT of the two DINT array you defined when you created the alarm trigger `::{PAC1}Program:Blending.AlarmArrayDint[0],L2` with the L2 modifier in the **Alarm Setup** dialog. The `::{PAC1}Program:Blending.AlarmArrayDint[1]/4` tag address is actually the 37th bit in the two DINT array. As a result the **Dint[1] bit 4 in alarm** message will appear in the **Alarm List** object when this button is pressed.

Note that you may have to hold the button down until the Alarm system sees the tags value change. The system response time depends on the network speed and the **Maximum update rate** set in the Alarm Setup.

The **Dint[1] bit 4 in alarm** alarm message will display in both the Alarm List and the pop-up alarm display.

The alarm is triggered with a momentary button. When the bit turns to one, the alarm is triggered and brings up the Alarm banner. As the alarm returns to a normal state when the button is released, the alarm banner will then be blank, as the alarm has returned to normal.

11. Click the **Close** button to close the pop-up alarm display.

12. Click the **Main** button when you are done with the **Alarming** display to return to the **Main** display.

13. Click on the **Shutdown** button to stop the test application.
Creating and Configuring Information messages

In this section you will use an *Information message* to alert operators that the tank is 75% full and will be ready to discharge soon.

1. Double-click *Information Messages* in the *Information* section of the Explorer Window

2. In the first line, type ‘1’ in the Trigger Value and type ‘Tank 1 level 75% full – Prepare to start discharge procedure’

   Repeat this step to create two more line entries for Tank 2 and Tank 3. Your Information message configuration should look like the following:

   ![Information Message Configuration](image)

   You can use copy / paste to fill out the Message tabs. Double click to select the message and use Ctrl – C to copy the message. Put your cursor in a new message row and use Ctrl – V to paste it.

3. Click the *Close* button; select *Yes* to save changes.
4. Enter "Tank Messages" in the Component name dialog box; select OK to save the message.

5. Select Information Setup in the Information section of the Explorer Window.

6. In the Message file name use the browse button and select the message file you created, Tank Messages.

7. In the Value field of the Connection section use the Tag browse button and select Program.Blending.TankLevels.


9. Press OK to save the changes.
The display [INFORMATION] will now open when a message is triggered by a non-zero value in the tag assigned to the Value connection. Let’s take a look at the default [INFORMATION] graphic. Double-click on the [INFORMATION] display in the project explorer.

The display which is selected in Information Setup must be configured as an On Top > Cannot Be Replaced display type. You can use the default display, [INFORMATION] or create your own. There are only two types of Information objects,

- The Information Message Display and
- The Acknowledge button

6. Close the [INFORMATION] display

Information messages and Alarms are checked by default in the Startup settings. Feel free to open the Startup settings to verify this.

Testing Information Messages

You can test the Information Messages section of the application now on the desktop, or wait until the lab is complete to test all components together on the PanelView Plus.

1. Select Test Application icon from the toolbar.

2. Leave the Main display open for a minute or so. You will see the Information message appear on top of the MAIN display. It will eventually close as the state of the tank level is controlled by timers to simulate the tank filling.

3. Select the Blending display – and wait for the Information message to appear on this display. Information and Alarm banner On top displays will open over any other displays so the operators can be alerted to a task or alarm at any time.
4. Press the **Ack [F1]** to acknowledge and close the *Information* display.

5. Return to the *MAIN* display and use the *Shutdown* button to stop the Test Application.
Creating and Configuring Recipes with the RecipePlus system

In this section you will learn how to

- Import and modify recipes model and understand the principles of Recipes.
- Create a recipe screen using the standard Recipe Plus components

Your factory is currently running 3 different types of Soda drinks. The sales of the Grapefruit soda are going so well that they want a separate blending line to make this soda drink.

The company has invested in a new blending line but they want to re-use the recipe from the original blending line.

Why do you need to use recipes?

A recipe is a set of numeric and string data values (ingredients) that can be downloaded /uploaded to their associated tags in a controller. Each ingredient has a pre-set data value assigned to it.

The set of data values for all the ingredients in a recipe is called a data set.

The set of numeric and string tags assigned to the ingredients in the recipe is called a tag set.

The ingredients, data sets, and tag sets are stored together in a recipe file.

You can create different pairs of data sets and tag sets for the same set of ingredients. Each pairing of data set with tag set is called a unit. Each unit is like a unique recipe.

At run time, the operator can select the unit (recipe) that applies to the current operation.

For example: A factory making soda could use the same ingredients and tag sets (tags in the controller), but depending on the flavor desired, could use different data sets to specify to use of certain flavor extracts. As another example, you might want to have multiple blending lines making the same soda. In this case, the data set for all the production lines will be the same, but the tags receiving the recipe information would be different for each production line. Units allow you to combine different tag sets and data sets for the same set of ingredients.
As you can see in this example there are 3 production lines. Line 1 is using its own controller (PAC1), Line 2 and 3 are using the same controller (PAC2) but different tags.

Each line has its own set of tags which are grouped in a Tag Set. Each tag in the tag set is linked to an ingredient.

In the recipe there are 3 different Data sets of soda flavors: Blueberry, Grapefruit and Kiwi.

Every data type within the data set is linked to an ingredient.

These three data sets can be used by all the production lines because we use a Unit to combine a Tag Set and a Data Set.

The FactoryTalk® View RecipePlus system overview:
Each recipe file can contain:
- 15,000 ingredients
- 500 data sets
- 50 tag sets
- 2,500 units (Combination of the data sets and tag sets)

When using 1 production line (1 tag set) the maximum number of units is 500.
Importing RecipePlus tags

Recipes have two tag settings that are used to determine the results of various recipe operations during run time.

The first tag is called the **Status Tag** and it shows the results of download, upload, upload and create, delete, rename, restore and save operations. The result status of these operations can be ‘start’, ‘successful’ or ‘with errors’.

The second tag is called the **Percent Complete Tag**. The recipe system writes to this tag during any recipe operation to show what percent of ingredients currently being processed have been completed.

Before we can have a look at these tags we are going to import them.

1. From the **Tools** menu select the **Tag Import and Export Wizard…** option.

2. Select **Import FactoryTalk View tag CSV files** from the drop down list.

3. Click **Next**.
4. To the right of the Project field, select the browse button.

5. Browse to C:\Users\Public\Public Documents\RSView Enterprise\ME\HMI Projects\Beverage_ME and select Beverage_ME.med.

6. Click Open, you just selected which ME application you want to import the tags to.

7. Click Next and click on the browse button, now you are going to select the file that contains the tags that you are going to import. The file is located at: C:\Lab Files\View ME – Fundamentals\ViewME

8. Select the Recipe_Tags.csv file and click Open.
9. Once the CSV file has been selected click *Next*.

![Tag Import and Export Wizard dialog box]

You will be presented with an import options dialog.

10. In this case you can just leave it as default and click *Next*.

![Import options dialog box]

*Skip existing:*

Means that tags in the import file that are also in the database will not be imported.

*Update existing:*

Means that the duplicate tags in the database will be updated with the info from the import file.

**Important:** The import wizard will not delete any tags. If you want to remove tags you need to do that in the tag database builder itself.

11. You will be presented with an overview of the source and destination files. Click *Finish* to import the tags into the application.

You will see a dialog box showing the status of the import
12. Click the X in the upper right corner of this dialog to close it.

About the tags you imported:

**HMI tag database**

In the HMI tag database, you can define data you want FactoryTalk View to monitor. There is one HMI tag database per HMI server. Each entry in the database is called an HMI tag. An HMI tag is a logical name for a variable in a device or in local memory (RAM) that is referenced in the HMI tag database.

All parts of the system use tag values. Graphics uses tag values to control the animation in a display or update a trend graph; Alarming monitors HMI tag values and compares them to "acceptable" limits; Data Log stores tag values to create a historical record.

HMI tags are required when you need extra information included with the tag value. This extra information can be a minimum/maximum value range or value units. If you do not need this information, you can use direct references to tags in other data servers.

**HMI tag types**

You can create three types of HMI tags, based on the data that they can store:

- Analog tags - store a range of values. These tags can represent variable states such as temperature or the position of rotary controls.
- Digital tags - store two states. Zero will be written as False and any non-zero number will be written as True. Use digital tags to represent devices that can only be on or off, such as switches, contacts and relays.
- String tags - store an ASCII string, series of characters, or whole words (maximum 82 characters). These tags can represent devices that use text, such as a bar code scanner which uses an alphanumeric product code.

**HMI tags you do not create**

A set of system tags is created automatically when you create an HMI server project, and stored in a folder called 'system' in the tag database.

System tags - store information generated while the system is running, including alarm reset time and date, system time and date, and current user information. There are analog, digital, and string system tags. You can use them anywhere you use other types of tags, but you cannot edit system tags.

**Data sources**

When you create an HMI tag you must specify what its data source will be at run time. Data source can be:

- **Device** - receives its data from a programmable controller through a direct driver or an OPC server. It can also receive data from another Windows program through an OPC server.
- **Memory** - data comes only from the value table rather than from a programmable controller or another program.
- **System** - the system tag is created by the system and stored in a folder called System.
Checking the imported recipe tags

Now you can go and verify the tags that you imported.

1. Go to the HMI Tags folder, expand it and double-click on Tags .
   As you can see the 2 tags are already created for you.

2. Close the Tags dialog.

![Tags dialog]

StatusTag
The status tag shows the results of recipe download and upload operations. The following table indicates the value of the status tag (in hexadecimal) when certain operations are performed:

<table>
<thead>
<tr>
<th>Operation</th>
<th>Start</th>
<th>Successful</th>
<th>With Errors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Download</td>
<td>0x01</td>
<td>0x02</td>
<td>0x04</td>
</tr>
<tr>
<td>Upload</td>
<td>0x011</td>
<td>0x012</td>
<td>0x014</td>
</tr>
<tr>
<td>Upload and Create</td>
<td>0x021</td>
<td>0x022</td>
<td>0x024</td>
</tr>
<tr>
<td>Delete</td>
<td>0x041</td>
<td>0x042</td>
<td>0x044</td>
</tr>
<tr>
<td>Rename</td>
<td>0x081</td>
<td>0x082</td>
<td>0x084</td>
</tr>
<tr>
<td>Restore</td>
<td>0x101</td>
<td>0x102</td>
<td>0x104</td>
</tr>
<tr>
<td>Save</td>
<td>0x201</td>
<td>0x202</td>
<td>0x204</td>
</tr>
</tbody>
</table>

PercentCompleteTag
The recipe system writes to this tag during any recipe operation to show what percentage of ingredients currently being processed have been completed.
Configuring the RecipePlus components

The RecipePlus feature allows users to read or write a set of values to a set of tags in a single operation. RecipePlus can also be used to compare two recipes to determine their differences.

The RecipePlus feature consists of several components:

- **RecipePlus Setup** – used to specify whether recipe files will be a part of the HMI project, and where the files will be stored for retrieval at run time.
- **RecipePlus Editor** – used to create and edit the recipe files by specifying the ingredients of the recipe, the data sets, the tag sets and the recipe units. The editor can also be used to compare data sets and tag sets within a single recipe.
- **RecipePlus table object** – used to display, modify and save the contents of a recipe at run time. The table also shows the differences between the recipe data values and the tags associated with the recipe (i.e. data set and tag set).
- **RecipePlus button object** – used to initiate recipe operations at run time (i.e. download, upload, upload and create, save, restore, delete and rename).
- **RecipePlus selector object** – used to select from a list of created recipes.

Importing the Recipe into your application

Since we are starting from an existing Blending Line we are going to import the recipe file.

1. Right-click on *RecipePlus Editor* and select *Add Component Into Application*..

![Add Component Into Application]

2. Navigate to the *C:\Lab Files\View ME - Fundamental\ViewME*.

3. Select the *Soda recipe.rpp* file and click *Open*.

![Select File]
The recipe has now been imported into the application.

Setting up RecipePlus

1. Under select . Make sure that the is selected.

   Make sure that the Recipe files are part of the HMI project radio button is selected.
At run time, recipe files can either be a part of the HMI project, or can be stored in some other location. Use the **RecipePlus Setup** dialog to specify where the recipe files will be located at run time.

At design time, only recipe files that are within the HMI project can be edited.

**Important**: If you choose to store the recipe files outside of the HMI project at run time, you must move the files manually from within the HMI project to the desired location before running the application.

**Recipe file location**

*Recipe files are part of the HMI project*

Select this radio button if recipe files will be part of the HMI project at run time. Recipe files appear under the **RecipePlus Editor** node in the Application Explorer. The recipe folder in the HMI project will be named RecipePlus.

When this option is selected, all of the recipe files in the HMI project will be included in the ME runtime application (*.mer). This is the default behavior.

*Recipe files are NOT part of the HMI project*

Select this radio button to specify an alternative location for recipe files at run time.

Click the browse button to browse to the desired location. At run time, the recipe system will look for recipe files in this folder rather than in the HMI project recipe folder.

2. Click **OK** to close this screen.

**Adding a TagSet to a Recipe**

1. Under **RecipePlus Editor**, double click **Soda Recipe**. This is the recipe file you just imported.

You will see the **General** tab as shown below.
General Tab:

*Runtime recipe name*
Type a unique, user-friendly name for the recipe file. This is the name the recipe selector will use at run time.

*Status tag*
Click the browse button to open the Tag Browser and specify a tag to be used as the status tag.

*Percent complete tag*
Click the browse button to open the Tag Browser and specify a tag to be used as the percent complete tag.

The Status tag and Percent complete tag are not filled in.

2. Click the `browse` button to the right of the *Status Tag* field, this will open the tag browser.

3. Highlight the `Beverage_ME` folder and select the `StatusTag`, click **OK**.
4. Repeat step 2 and 3 above for the **Percentage Complete Tag** field and this time select the **PercentCompleteTag** tag.

5. Click on the **Ingredients** tab.
This is a list of all the ingredients that are needed to create *Blueberry, grapefruit* or *kiwi*.

As you can see we only have 1 tag set which is called *Tag Set 1*.

If you click on the *Units* tab, you will see that the units are all linked to the *Tag Set 1*. Now go back to the *Ingredients* tab.

6. Select the *Recipe* menu option located between *Edit* and *View*. 

![Image of FactoryTalk View Studio - Machine Edition]
7. In the pull down menu select *Rename Tag Set*...

8. Enter ‘Blending Line 1’ in the *Enter new tag set name:* field for *Tag Set 1*.

9. Click **OK** to close the **Rename Tag Set** dialog.
   
   If you now click on the **Units** tab you will see that the *tag set* of the units is also modified.

10. Select the **Ingredients** tab again.

11. In the **Recipe** pull down menu select **Insert Tag Set**....

12. Enter ‘Blending Line 2’ in the **Enter the name of the new tag set:** field.
13. Click **OK**.

For this new **Blending Line 2** we need to fill in the associated tags using the tag browser.

14. To open the **Tag Browser**, select the top cell under the **(Tag Set) Blending Line 2** column heading, right-click in the cell and select **Tag Browser**, or select **Tag Browser** from the Edit menu.

You will need to change the column widths using the mouse in order to see the last column.

15. Fill in the tags for the **Blending Line 2** as shown below:

Expand **Recipe** and select the **Flavor** folder in the left hand pane of the tag browser for the **Flavor** ingredient tag.

<table>
<thead>
<tr>
<th>Ingredient</th>
<th>Tag Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>Water</td>
<td>{{::[PAC1]Program:BlendingL2.Recipe.Water}}</td>
</tr>
<tr>
<td>Syrup</td>
<td>{{::[PAC1]Program:BlendingL2.Recipe.Syrup}}</td>
</tr>
<tr>
<td>Acid</td>
<td>{{::[PAC1]Program:BlendingL2.Recipe.Acid}}</td>
</tr>
<tr>
<td>Carbon Dioxide</td>
<td>{{::[PAC1]Program:BlendingL2.Recipe.CarbonDioxide}}</td>
</tr>
<tr>
<td>Flavor</td>
<td>{{::[PAC1]Program:BlendingL2.Recipe.Flavor}}</td>
</tr>
</tbody>
</table>
Your Tag Set 2 column should look like this:

<table>
<thead>
<tr>
<th>(Tag Set)</th>
<th>(Tag Set)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blending Line 1</td>
<td>Blending Line 2</td>
</tr>
</tbody>
</table>

In this case we are using the same controller but a different routine in the controller for the second blending line. The tag set could also come from a different controller.

18. Now select the Units tab.
In this lab, Blending line 1 is making 3 different soda drinks Blueberry, Grapefruit and Kiwi. Now you will create another unit for Line 2 using the Grapefruit dataset.

19. Select Insert and enter ‘Grapefruit Line 2’ in the Unit name field, use the drop down list buttons and select Grapefruit and Blending Line 2 as shown below.

![Insert Unit dialog]

20. Select OK to close the Unit dialog.

21. Select Close in the RecipePlus Editor, and select Yes to save the recipe.

Now the unit is created so that it can re-use all the values used in the original line recipe for Grapefruit on the new Blending Line 2.
Creating the recipe display using standard library components

In this section we are going to create a recipe screen based on the standard library components.

1. Expand Libraries and then double-click RecipePlus_Components.

2. Once the screen is open either go to the Edit menu and select Select All or press <CTRL> + A to select all components on the library screen.

3. Either go to the Edit menu and select Copy or press Ctrl + C to copy all the components that are selected.

4. Go to Graphic > Displays option, right mouse click and select New.

5. Click on the new screen and press Ctrl + V to paste the library components on this new display.

The recipe screen now contains all the necessary components to manage recipes. The only thing you need to do now is to create a button called Back that will bring you to the screen that is opened before the Recipe screen.

6. Go to the Objects menu option and select Display Navigation and select Return To.

7. Draw a rectangle in the open space above the Recipe Status.

8. Double-click on the button and on the Label tab enter "Back" in the Caption field.

9. Click on OK to close this property dialog.
10. Close this display and click **Yes** when prompted to save your changes.

11. Enter ‘Recipe’ in the **Component name** field and click **OK**.

12. Close the **RecipePlus_Components** library display. Select **No** when prompted to save the display.

**Congratulations!!** You have successfully configured Recipe Plus, created Recipe Plus objects on a display and configured display navigation.
Testing Recipes

You can test the Recipe section of the application now on the desktop, or wait until the lab is complete to test all components together on the PanelView Plus.

1. Select **Test Application** icon from the toolbar.

2. On the **Main** display click the **Recipe Goto** button to navigate to the **Recipe** display

Explanation of the recipe display:

**Recipe Selection:**

The RecipePlus selector is a list of recipe files and recipe units available in the application. Use the selector to select from the list of recipes and their associated units. When a recipe is selected, it becomes the ‘current’ recipe, and becomes the object of subsequent recipe operations, such as upload, download, restore, etc., performed on that display.

When the selector object is selected, the operator can scroll through the list and select recipe files using key button objects, or by using the arrow keys and the Enter key on a keypad or external keyboard. Key button objects can also be linked directly to the selector so that they will work whether the selector is selected or not.

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recipe</td>
<td>Displays the Name of the recipe as shown in the General tab of the Soda Recipe. Right above the Status Tag.</td>
</tr>
<tr>
<td>Unit</td>
<td>Displays all the units that have been created in the Soda Recipe.</td>
</tr>
</tbody>
</table>

**Recipe Table:**

The RecipePlus table object lists the ingredients in the recipe file most recently selected in the RecipePlus selector and restored. It can also be used to edit the data values for ingredients and to save those edited values to the recipe file.

The explanation for what each column in a **RecipePlus Table** represents at runtime is summarized below:

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ingredient</td>
<td>Displays the name of each ingredient in the recipe.</td>
</tr>
<tr>
<td>Current</td>
<td>Displays the current value of each ingredient in the tag associated with the ingredient.</td>
</tr>
<tr>
<td>Recipe</td>
<td>Displays the data value assigned to each ingredient. This data value can be modified by selecting the ingredient row in the table and pressing Enter.</td>
</tr>
<tr>
<td>Compare</td>
<td>If the tag value and the recipe data value differ, an X appears in this column to alert the operator of the difference.</td>
</tr>
</tbody>
</table>
Tag Name | Displays the tag name associated with each ingredient.
--- | ---

In the status bar at the bottom of the table the recipe file name, the recipe unit name, the number of visible ingredients and the total number of ingredients are shown.

**Recipe Buttons:**
The RecipePlus system has a button with seven different action properties. The button types and their functionality descriptions are summarized in the table below:

<table>
<thead>
<tr>
<th>Button Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Download</td>
<td>Writes the ingredient values in the data set of the recipe unit currently selected in the selector object to the tags in the unit's tag set.</td>
</tr>
<tr>
<td>Upload</td>
<td>Reads the tag values in the tag set of the recipe unit currently selected in the selector object and saves those values to the unit's data set.</td>
</tr>
<tr>
<td>Upload and Create</td>
<td>Reads the tag values in the tag set of the unit currently selected in the selector object and creates a new recipe unit by writing those values to a new data set.</td>
</tr>
<tr>
<td>Save</td>
<td>Saves the data value changes made in the Table back to the recipe file.</td>
</tr>
<tr>
<td>Restore</td>
<td>Reads the recipe file and displays it in the table. Viewing the recipe in the table will show the user the data values in the recipe, the data values currently in the controller and the tag names used.</td>
</tr>
<tr>
<td>Delete</td>
<td>Deletes the recipe unit currently selected in the selector object.</td>
</tr>
<tr>
<td>Rename</td>
<td>Renames the recipe unit currently selected in the selector object.</td>
</tr>
</tbody>
</table>

**Scenario 1:**
The operator wants to see if the Kiwi recipe is in the Controller.

**Howto:**
1. Select the **Kiwi** recipe in the **Recipe Selection** using the arrow keys.
2. Click on the **Restore** button.
   - For every ingredient that is different in the controller compared to the Kiwi Recipe the column ‘Compare’ will show an ‘X’

**Scenario 2:**
The operator wants to check the Grapefruit recipe, download it and check if it’s in the controller

**Howto:**
1. Select the **Grapefruit** recipe in the **Recipe Selection** using the arrow keys.
   - Optional: Click on the ‘Restore’ button this way you can see what’s in the controller
2. Click on the **Download** button and check that the status shows ‘**Download Successful**’.

3. Now click again on **Restore** and check that there is no ‘X’ shown in the **Compare** Column

4. Use the **Back** button to return to the **MAIN** display and select the **Shutdown** button to close the test display.

---

To view the recipe file without FTView Machine Edition you can use a free tool called RSView Enterprise File Viewer. The tool can be found in the Rockwell Knowledge Base ([http://www.rockwellautomation.com/knowledgebase/](http://www.rockwellautomation.com/knowledgebase/))

The ID for the tool is 33924 - RSView Enterprise File Viewer Utility

The RSView Enterprise File Viewer utility can be used to read and save the following files:

- SE Datalog file (*.DAT, *.OBF)
- ME Datalog file (*.LOG)
- ME Alarm Log file (HISTORY.ALM)
- dBASE file (*.DBF)
- MERecipe (*.RPP)

The SE Datalog files, ME Datalog files, ME Recipe files and ME Alarm Log file can be saved as CSV or DBF files.

In the tool click Open, then change the ‘Files of Type’ to RecipePlus File (*.RPP)
Understanding the power of Global Object Parameters

In this section you will learn how to

- Create global objects with parameter definitions
- Use placeholders in global objects
- Understand nested parameters

Reviewing Basic Parameter Passing

Let's review how basic parameters work.

When you create a graphic display that has tag placeholders, you can specify a parameter list that will supply tag or folder names, or parts of tag names, for the tag placeholders. A placeholder is a cross-hatch character (#) followed by a number from 1 to 500. At run time, you can use a parameter list to replace the tag placeholders with whatever string you have defined for the placeholder.

The first tag in the list will replace #1 in the display expressions, the second tag in the list will replace #2, and so on. A parameter list stays with the object when it is copied or duplicated.

Global Object Parameters

When importing a single display from xml, you must have a graphic file to import the xml components to. As you don't have a graphic file to import the global object display to, we are going to use a multiple displays batch file.

Your customer wants his operators to see the status of all three tanks on the same display. In this section, you will create one faceplate as a global object, and create three instances of the faceplate on the Blending display.

When importing a single display from xml, you must have a graphic file to import the xml components to. As you don’t have a graphic file to import the global object display to, we are going to use a multiple displays batch file.

1. Right click on Graphics > Global Objects and select Import and Export.
2. Select the *Import graphic information into displays* and then select *Next*.

3. Select the *No* radio button when you are prompted to backup the displays that will be modified, then select *Next*.
4. Choose the *Multiple displays batch import file*, then select **Next**

![Graphics Import Export Wizard - Import File Type](image)

5. Use the **browse ellipsis** button to browse to the `C:\Lab Files\View ME – Fundamental\ViewME` folder and select **BatchImport_Global_Beverage_ME.xml**, then click **Open**

How was the xml file generated?

When you export graphic displays, an XML file with the same name as the display and with the file extension `.xml` is created for each display exported. In addition, a batch import file is created that is given the name `BatchImport_<applicationname>.xml`.

Using this batch file allows you to import single or multiple XML files to existing or new displays.
6. Leave the default radio button *Create new objects on the display* and select *Finish*

![Graphics Import Export Wizard - Multiple Import File](image)

7. Close the *DisplaysImport.txt* file.

8. Open the *Tank.GO* from *Graphics > Global Objects* in the Application Explorer.

![Tank.GO](image)

The faceplate has three numeric display objects. We will configure tag placeholders that reference two base tags.

*When using Global object parameters, it's good practice to use partial tag substitution.*

9. Double-click on the first numeric display beside *Tank Level (L):*
10. Select the \textit{Connections} tab. Use the \textit{browse ellipsis} button to browse to \texttt{Program:Blending.Tank1.Level} and select \texttt{OK}.

11. Click the \textit{Expression} ellipsis button to open the \textit{Expression Editor} which allows easy editing of the connection.

12. Modify the expression to use a placeholder of \#1, as shown below, \texttt{(#1.Level2}}, to represent the tank number when the global object is copied to the \textit{Blending} display.
13. Select OK to close the expression editor, then select OK to close the Numeric Display Properties.

There are three tanks on the Blending display, each tank has a set of values associated with it in the SoftLogix Controller. These values will be assigned when the global object is copied to the Blending display.

![Image of tanks]

You will repeat these steps to assign connections to the other numeric display and string objects.

14. Double-click on the second numeric display beside Water Ratio:

15. Select the Connections tab. Use the browse ellipsis button to browse to Program:Blending.Recipe.Water and select OK

![Image of numeric display properties]

16. Modify the expression to use a placeholder of #2 to represent the recipe tags when the global object is copied to the Blending display.

![Image of numeric display properties]

17. Select OK to close the expression editor (if used), then select OK to close the Numeric Display Properties

18. Double-click on the third numeric display beside Syrup Ratio:

19. Select the Connections tab. Use the browse ellipsis button to browse to Program:Blending.Recipe.Syrup and select OK
20. Modify the expression to use a placeholder of #2 to represent the recipe tags when the global object is copied to the **Blending** display.

![Numeric Display Properties](image)

21. Select **OK** to close the expression editor (if used), then select **OK** to close the **Numeric Display Properties**

22. Let's take a look at the String display object. Double-click on the **s Information** text object.

![s Information](image)

Tag placeholders can be used as variables in text objects. This provides a method to display a descriptive name of the base tag used in the faceplate.

![Text Properties](image)

The syntax for embedded variables is:

- an opening escape sequence ‘/*’
- followed by the letter ‘S’
- followed by a colon ‘:’
- followed by a numeric length field (if the Fixed number of characters check box is checked in the String Variable dialog box)
- followed by a space character
- followed by the tag name, tag placeholder (#1, #2, etc.) or combination
- followed by a closing escape sequence ‘*/’
- followed by static text

This has been configured for you.
23. Select **OK** to close the *Text Properites*.

**Assigning Global Object Definitions**

Now that the display objects have been configured on the faceplate, you will assign global object parameters to the faceplate.

> It’s good practice to group all the components in a global object so that you configure the Global Object definition once.

1. Group the numeric display, string display and panel objects by using the select tool.

2. Use the **Group** icon to group the objects. Your object should look like the picture on the right.

3. Right-click on the newly grouped item, and select **Global Object Parameter Definitions**.
4. In the dialog box, you will define two tag placeholders. Enter ‘#1’ in row one. Enter ‘Tank Information’ in the Description field. Enter ‘#2’ in row two. Enter ‘Recipe Information’ in the Description field.

5. Select OK to close the dialog.

It’s time to save the global object display.

6. Select the Save icon to save the Tank.GO global object display.

Now you will copy three instances of the global object to the Blending display.

7. Select the grouped global object, right click and select Copy from the menu.

8. Select the Blending display from Graphics > Displays in the Explorer Window.

Note: Your application may look slightly different if you completed previous sections such as Alarms.

9. Double-click in the display to bring up the context menu and select Paste.

10. Move the faceplate to the tank on the left.

11. Right-click on the pasted global object and select Global Object Parameter Values.
12. Select the **tag browse ellipsis** in row two in the dialog. Browse for *Program:Blending.Tank1* base tag.

During runtime, the **Global Object Parameter Value** selected here will be used to replace the #1 tag placeholder you configured in **Global Object Parameter Definition**.

13. Select **OK** to accept the tag.


15. Select **OK** to accept the tag. Your **Global Object Parameter Value** should look like the following.

16. Select **OK** to close the **Global Object Parameter Value** dialog window.

You will configure two more instances of the Global Object for the other two tanks. One option is to paste another instance of the global object from the Tank_GO global object display. As the tags are similar, you can save time by copying the instance of the global object that you configured above.
17. Right-click the global object on the **Blending** display, and select **Copy** from the menu. Right-click in the display and select **Paste**. Move the new copy to the second tank.

18. Click on the pasted global object and select **Global Object Parameter Values**.

```
Modify the **Global Object Parameter Values** as follows:
#1 = ‘::[PAC1]Program:Blending.Tank2’
#2 = ‘::[PAC1]Program:Blending.Recipe’
```

19. Select **OK** to close the **Global Object Parameter Value** dialog window.

20. Repeat the steps tank 3.

```
Modify the **Global Object Parameter Values** as follows:
#1 = ‘::[PAC1]Program:Blending.Tank3’
#2 = ‘::[PAC1]Program:Blending.Recipe’
```

Your display should look similar to the following except there won’t be any live data:
21. Select Save \(\text{Save}\) from the menu.

Congratulations! You’ve completed the Global Object Parameters section.

Now you have completed all the sections in this lab, we’ll create a runtime file and download it to your PanelView Plus.
Create a runtime (MER) file, download and start the application

You've created all the components you need to monitor the Blending area of the Beverage plant. You are ready to create a runtime file and download it to the PanelView Plus 6!

Create the MER runtime file

1. Select the Application > Create Runtime Application menu item.

Note that the top level Studio menu options can change depending on what application components you are working on however the Create Runtime Application menu option will always be located under Application.

2. Using the Create Runtime Application dialog save the runtime project using the default name of the application by clicking the Save button.
The **Conversion to development application** options enable you to specify if a runtime application can be restored to the design files project using the Application Manager.

**Always allow conversion [Default]**
The design information is always included with the runtime, so that it may be restored from the MER.
The resulting MER requires more terminal memory to store the file.

**Never allow conversion**
Design information cannot be recovered from an MER created with this option selected.
The MER created requires the least amount of terminal memory.

**Conversion protected by password**
When using Application Manager to extract the design information from the runtime file, the user will prompted for the configured password.
The resulting MER requires more terminal memory to store the file.

While FactoryTalk View Studio for Machine Edition is creating the runtime MER file a progress dialog will appear.

After creating the runtime MER file, the FactoryTalk Studio for Machine Edition continues to display the open application.

**Download a runtime (MER) file to a PanelView Plus 6 terminal**

1. To download the runtime MER to the PanelView Plus 1250 terminal at your workstation, first, select the *Tools* > *Transfer Utility* menu item
The **Transfer Utility** will open.

2. Click the **Source File** browse button to select the runtime MER file to download.

This opens the **Select File to Download** dialog.
3. Click **Beverage_ME.mer** to select the runtime file. The **File name** box should update to reflect the selection.

4. Click the **Open** button to complete the project selection.

5. Double-click the **EtherNet, Ethernet** driver to expand the network view.

6. Select the **192.168.1.20, PanelView Plus 1250, PanelView Plus 1250** item by clicking on it once.

7. To initiate the download process, click the **Download** button. Select **Yes** if you are prompted to overwrite the destination file.

   During the download a progress dialog will update. An example is shown below:
When the download process completes a confirmation dialog is shown.

8. Click the OK button to acknowledge the dialog.

9. Click the Exit button to close the Transfer Utility.

Running an application on a PanelView Plus terminal

In this lab section you will interact with the PanelView Plus 6 terminal. Execute the following steps to run the FactoryTalk Machine Edition application you just completed.

1. Double-tap the FTViewME Station icon on the PanelView Plus desktop to open FactoryTalk View ME Station.

2. Press the Load Application [F1] button.

If you cannot see the FactoryTalk Machine Edition Station application display, please ask for assistance.
3. Select the *Beverage_ME.mer* file from those available from the terminal’s *Internal Storage*.

![Image of file selection dialog]

4. Press *Load [F2]* to load the runtime file into memory.

5. When prompted, press *Yes [F7]* to overwrite the terminal’s current communication configuration with the configuration contained within the *Beverage_ME.MER* file.

6. Once successfully loaded, press the *Run Application [F2]* to start executing the runtime file.

![Image of run application dialog]

While the terminal is starting the application, an update dialog is displayed.

```
Starting application.
Please wait...
```
Your display should look similar to the one above. You may see some Alarm message on the PanelView Plus; if you do, press the Close button on the Alarm window.

Let’s test the **Blending** faceplates.

1. Click on the **Blending** icon to open the **Blending** display

Note the different values in each faceplate, even though they used the same global object! Feel free to review the other displays if you skipped the Test Application sections previously.

Please shutdown the application when you are finished reviewing the application. Comments? Please feel free to ask questions or share your comments with the lab moderators.
Exploring FactoryTalk View Studio for Machine Edition Interface

The FactoryTalk View Studio for Machine Edition Application Windows is divided into several key elements:

**Application Menu**
- Used to interact with the application;
- Open/Close/Create new applications;
- Import/Export information.
- The menu changes context based on what project object is open in the **Work Pane**.

**Explorer Pane**
- Contains all objects related to an application project.
- Application objects are opened in the **Work Pane**.
- The Explorer’s content is described in more detail on the next page.

**Diagnostic List**
- Contains status and error messages related to system, application and project.

**Objects Toolbar**
- Provides easy access to objects that are used on displays to create the user interaction. Ex. Numeric Input, String Display, Ramp button, etc.

**Graphics Toolbar**
- Provides easy access to tools that are used to manipulate objects on a display. Ex. Rotate, Group, Ungroup, etc.

**Work Pane**
- Open project object contents are displayed in this area.
- In this example, an untitled display has added to the application.
Now let's take a close look at the objects in the **Explorer Pane**.

- **Local (CORE)** contains computer-scoped components, FactoryTalk View Data Server and FactoryTalk Services Platform objects.

- **L15** contains application-scoped components like the FactoryTalk View HMI Server and RSLinx Enterprise configurations. The FactoryTalk View Data Server object is used to add additional data servers (Ex. OPC) to an application project. This is an advanced operation and not covered in the manual.

- **L15** contains all information scoped to the HMI server such as, Project Settings, HMI Tags, Graphic files, Alarms, Information Messages, Macros, Data Logs, and Recipes.

- **System** contains project-scoped settings such as Resolution, Security settings, Startup graphic files, Diagnostic configuration.

- **HMI Tags** contains all tags resident in the memory of the HMI Server. Applications for Logix Controllers use direct tag referencing eliminating the need to create HMI tags to communicate with the PLC.

- **Graphics** contains all graphic images used in the application as displays, or contained within displays and parameter files. Parameter files are one way to enable graphic reuse between applications.

- **Symbol Factory** provides access to thousands of stock images organized in an easy to use library.

- **Alarms** contains the application's alarm configuration including triggers and messages.

- **Information** contains the application's information message configuration.

- **Logic and Control** contains the application's macros.

- **Data Log** contains any Data Log Model configured for the application.

- **RecipePlus** contains the Recipe management system configuration and any configured recipes.

- **RSLinx Enterprise** contains communication shortcuts used by the application.

- **System** contains FactoryTalk Services Platform related configuration. This is primarily used to manage users and groups in a FactoryTalk Machine Edition application.
Lab VZ03 – FactoryTalk View Machine Edition – Project Skills
Lab Setup and Configuration Information

### Lab Information

<table>
<thead>
<tr>
<th>Lab Name</th>
<th>VZ03 - FactoryTalk View Machine Edition and PanelView Plus Project Skills Lab</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lab Description</td>
<td>This hands-on lab focuses on fundamental FactoryTalk Machine Edition concepts for beginning users. Topics include data collection, trending, global objects, global object parameters, alarming and information messages.</td>
</tr>
<tr>
<td>Lab Creator</td>
<td>Rhonda Stock</td>
</tr>
<tr>
<td>Date Created</td>
<td>03/25/2011</td>
</tr>
<tr>
<td>Updates:</td>
<td></td>
</tr>
<tr>
<td>04/25/2011</td>
<td>Bob Syms</td>
</tr>
<tr>
<td>04/17/2012</td>
<td>Wil Mattheis</td>
</tr>
</tbody>
</table>

### Hardware Configuration per Student

<table>
<thead>
<tr>
<th>Qty</th>
<th>Demo Cat.# / Description</th>
<th>Communication</th>
<th>Location</th>
<th>Firmware</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Computer workstation consisting of</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2 GHz CPU</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>4 GB RAM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Hard drive. Minimum 40GB capacity</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>USB Keyboard</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>USB Mouse</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Ethernet adapter.</td>
<td>Configured for DHCP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Ethernet adapter.</td>
<td>IP Address: 192.168.1.1, Subnet: 255.255.255.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Video adapter. Minimum resolution: 1260 x 1024 pixels</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>LCD Display. Minimum resolution: 1260 x 1024 pixels</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Ethernet CAT5E cross-over cable 5ft (used with PanelView Plus terminal)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Ethernet CAT5E cable 5ft (used with Event classroom network)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>PanelView Plus 6 1250 – build 6.10.17 or greater</td>
<td>IP Address: 192.168.1.20, Subnet 255.255.255.0</td>
<td></td>
<td>Firmware revision must support ViewPoint 2.10</td>
</tr>
</tbody>
</table>

---
### Computer Information

<table>
<thead>
<tr>
<th>Location</th>
<th>Files</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computer Name</td>
<td>Varies by machine</td>
</tr>
<tr>
<td>IP Address (NIC 1)</td>
<td>DHCP – connected to Event classroom network</td>
</tr>
<tr>
<td>IP Address (NIC 2)</td>
<td>Host computer TCP/IP setting: DHCP</td>
</tr>
<tr>
<td></td>
<td>VMWare image TCP/IP setting:</td>
</tr>
<tr>
<td></td>
<td>IP Addr: 192.168.1.1</td>
</tr>
<tr>
<td></td>
<td>Subnet: 255.255.255.0</td>
</tr>
<tr>
<td>Operating System</td>
<td>Windows 7 Professional</td>
</tr>
</tbody>
</table>

### Basic Setup Diagram

#### Student Station

PanelView Plus 6 1250

- **IP Addr**: 192.168.1.20
- **Subnet**: 255.255.255.0

- **CAT5E cross-over cable**

- **NIC 2**
  - Host computer TCP/IP setting: DHCP
  - VMWare image TCP/IP setting:
    - **IP Addr**: 192.168.1.1
    - **Subnet**: 255.255.255.0

- **NIC 1**
  - Host computer TCP/IP setting: DHCP

- **CAT5E cable**

To Classroom network

DHCP assigned IP
### Application/Programming

<table>
<thead>
<tr>
<th>Location</th>
<th>Files</th>
</tr>
</thead>
<tbody>
<tr>
<td>C:\Lab Files\ViewME – Fundamental\PVP</td>
<td>None Required</td>
</tr>
<tr>
<td>C:\Lab Files\ViewME – Advanced\SoftLogix</td>
<td>InstantFizz.acd (slot 2)</td>
</tr>
<tr>
<td>C:\Lab Files\ViewME – Advanced\ViewME</td>
<td>These files will be used in application creation</td>
</tr>
</tbody>
</table>

### Additional Equipment Required per Workstation

<table>
<thead>
<tr>
<th>Qty</th>
<th>Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Ethernet CAT5E cross-over cable 5ft (used with PanelView Plus terminal)</td>
</tr>
<tr>
<td>1</td>
<td>Ethernet CAT5E cable 5ft (used with Event classroom network)</td>
</tr>
<tr>
<td>1</td>
<td>PanelView Plus 6 1250, NOTE: ftp must be configured and running on the terminal with the default directory configured as blank and NOT \Temp.</td>
</tr>
</tbody>
</table>

### RSLinx - DDE/OPC Topic Configuration

<table>
<thead>
<tr>
<th>Topic Name</th>
<th>Path to Hardware</th>
</tr>
</thead>
<tbody>
<tr>
<td>N/A</td>
<td></td>
</tr>
</tbody>
</table>

### RSLinx - Driver Configuration

<table>
<thead>
<tr>
<th>Topic Name</th>
<th>Path to Hardware</th>
</tr>
</thead>
<tbody>
<tr>
<td>Workstation, CORE</td>
<td>RSLinx Gateways, Ethernet</td>
</tr>
<tr>
<td>Workstation, CORE</td>
<td>AB_VBP-1, 1709-A17/A Virtual Chassis</td>
</tr>
<tr>
<td>Workstation, CORE</td>
<td>00, Workstation, RSLinx Server</td>
</tr>
<tr>
<td>Workstation, CORE</td>
<td>01, RSLinx Enterprise - Desktop, RSLinx Enterprise - Desktop</td>
</tr>
<tr>
<td>Workstation, CORE</td>
<td>02, 1789-L60/A SoftLogix5860 Controller, InstantFizz</td>
</tr>
<tr>
<td>Workstation, CORE</td>
<td>03, 1789-L60/A SoftLogix5860 Controller, ME_Intro</td>
</tr>
<tr>
<td>Workstation, CORE</td>
<td>04, SoftLogix5800 EtherNet/IP, SoftLogix5800 EtherNet/IP</td>
</tr>
<tr>
<td>Workstation, CORE</td>
<td>06, 1789-L60/A SoftLogix5860 Controller, Block_Machine</td>
</tr>
<tr>
<td>AB_VBP-1 (no topic required)</td>
<td></td>
</tr>
</tbody>
</table>

### RSLinx Enterprise - Shortcut Configuration

<table>
<thead>
<tr>
<th>Shortcut Name</th>
<th>Path to Hardware</th>
</tr>
</thead>
<tbody>
<tr>
<td>PAC1</td>
<td>Ethernet, Ethernet &gt; 192.168.1.1 &gt; Backplane &gt; 2, 1789-L60 v19</td>
</tr>
</tbody>
</table>
## Application Versions

<table>
<thead>
<tr>
<th>Vendor</th>
<th>Software</th>
<th>Version</th>
<th>Service Pack</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rockwell</td>
<td>FactoryTalk Activation Manager</td>
<td>3.40</td>
<td>CPR9 SR4</td>
</tr>
<tr>
<td>Rockwell</td>
<td>FactoryTalk Diagnostics</td>
<td>2.50.00</td>
<td>CPR9 SR5</td>
</tr>
<tr>
<td>Rockwell</td>
<td>FactoryTalk Service Platform</td>
<td>2.50.00</td>
<td>CPR9 SR5</td>
</tr>
<tr>
<td>Rockwell</td>
<td>FactoryTalk View Machine Edition</td>
<td>6.10.00</td>
<td>CPR9 SR4</td>
</tr>
<tr>
<td>Rockwell</td>
<td>FactoryTalk ViewPoint ME</td>
<td>2.10.7</td>
<td></td>
</tr>
<tr>
<td>Rockwell</td>
<td>RSLinx Enterprise</td>
<td>5.40.0000</td>
<td>CPR9 SR4</td>
</tr>
<tr>
<td>Rockwell</td>
<td>RSLinx Classic</td>
<td>2.59.00</td>
<td>CPR9 SR5</td>
</tr>
<tr>
<td>Rockwell</td>
<td>RSLogix 5000</td>
<td>19.01.00</td>
<td>CPR9 SR3</td>
</tr>
<tr>
<td>Rockwell</td>
<td>SoftLogix 5800</td>
<td>19.01.00</td>
<td></td>
</tr>
<tr>
<td>Rockwell</td>
<td>Silverlight</td>
<td>4.0.60129.0</td>
<td></td>
</tr>
<tr>
<td>Rockwell</td>
<td>.NET Framework</td>
<td>1.1</td>
<td></td>
</tr>
<tr>
<td>Microsoft</td>
<td>Internet Explorer</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>Rockwell</td>
<td>PanelView Plus 6</td>
<td>6.10.17 or greater</td>
<td></td>
</tr>
</tbody>
</table>

## Required Pre-Lab Configuration

### Configure the Virtual Network

1. Select **Virtual Network Editor** from the **Edit** menu to open the **Virtual Network Editor** dialog.
2. The dialog below will appear.

3. Select the Host Virtual Network Mapping tab

4. For VMnet0 select the physical ethernet adapter that will be connected to the PanelView Plus terminal.

5. Click the OK button to commit the changes that have been made.