Planning Applications

Dan Kandray

January 30, 2020

Planning Steps

1. Understand the machines and processes you're automating
2. Plan Tag Database
3. Plan graphic displays and navigating between them
4. Plan languages
5. Plan alarms
6. Provide information for the operator
7. Plan trends
8. Plan recipes
9. Design a secure system

Understand the process

- Talk to machine operators and other experts who are going to use the system. Find out what information they need to optimize machine operations and processes.

- Talk to management and management information systems staff to find out what information they need to support planning decisions.

- Break up each section of the process into its constituent parts.

- Determine what type of communications you'll be using — which network types, data servers, and devices.

- Determine which process variables you need to have access to and identify their locations in the programmable controllers.
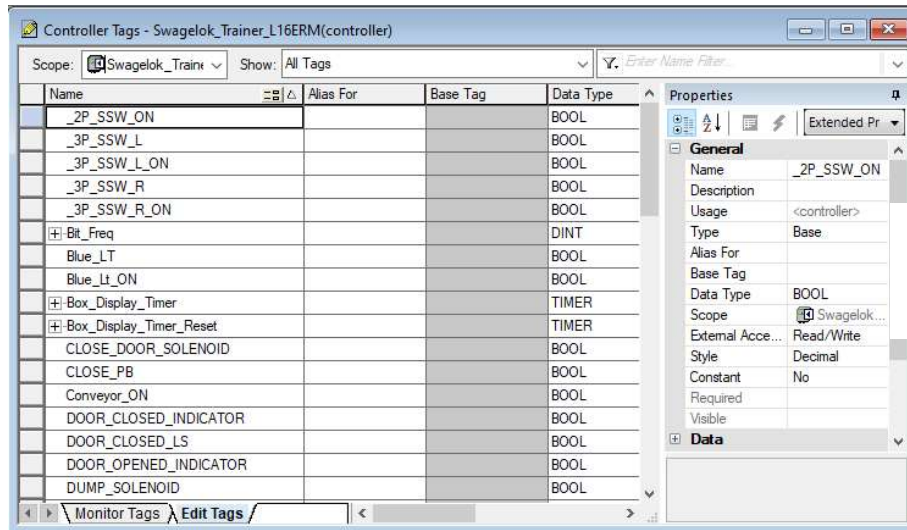
Collect information

- Flowcharts of your processes (or process and instrument diagrams).

- A list of programmable controller data table or register addresses your application will have access to.

- Machinery systems documentation.
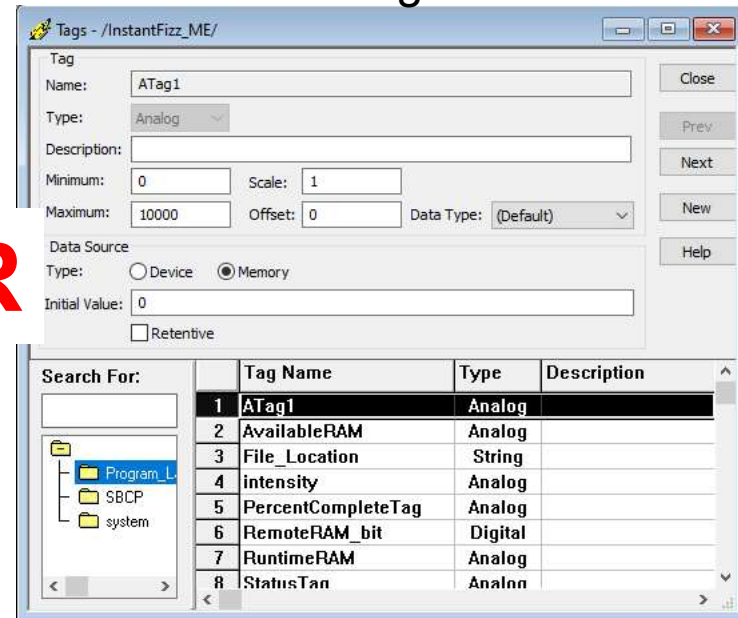
- Alarm requirements

Plan Tag Database

Options:

Data Server Tags (Direct Referencing)

Tags obtained directly from the data server (FactoryTalk Linx) – in other words, tabs obtained directly from PLC

OR

HMI tags

Tags you create in FactoryTalk View are called HMI tags

Data Server tags

- To obtain the best system performance for your application, use Data Server tags where possible. (Best Practice)

  - Data Server tags usually update faster than HMI tags.

  - Using data server tags allows you to add, modify, or delete tags in a device without having to duplicate the changes in the FactoryTalk View HMI tag database.

  - Some devices, for example, Logix 5000 processors, support complex data types, such as arrays and structures. Your controller might have structures that contain hundreds of member elements.
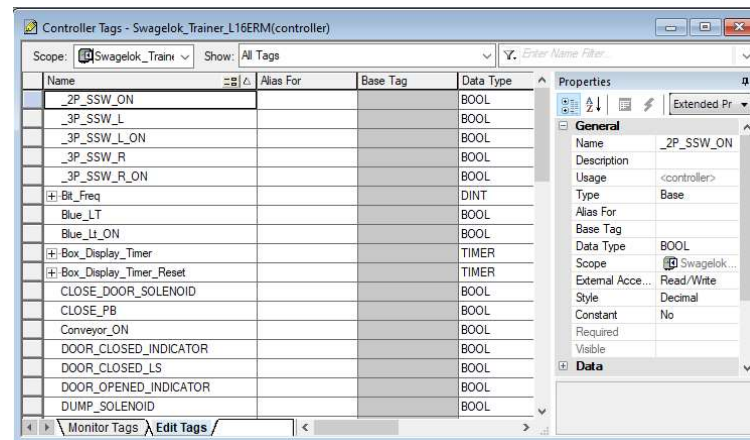
HMI tags

- Use HMI tags if the data server you are using does not provide for:
  - Scale or offset a value.
  - Set minimum or maximum limits on a value (that is, providing a range of values).
- Use HMI tags to manipulate tag data and to store tag values in the runtime computer's memory.
  - For example, you might need to store a value in memory:
    - To store the result of a calculation.
    - Temporarily, for example, a counter or index.
    - To maintain information about the system's current state, for example which graphic display was last opened.
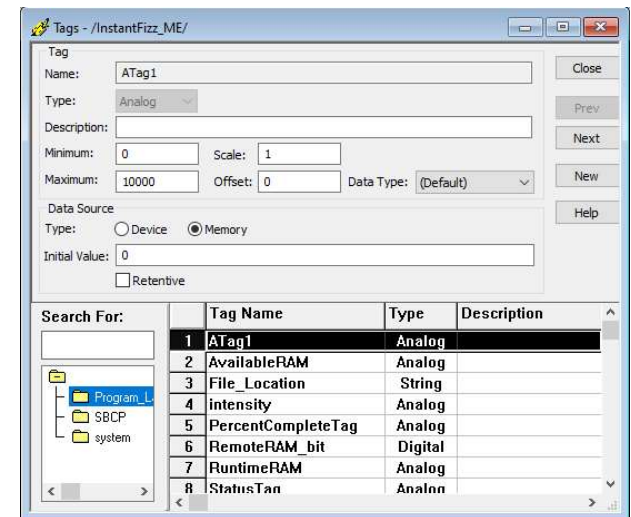
  CAUTION: Tag manipulation will be invisible to PLC troubleshooters

Plan graphic displays

- Determine the best way for users to navigate through your displays and develop a template to establish a common look and feel for your displays

- Develop a hierarchy of displays that provide progressively more detail as users move through them and plan how users will navigate through the hierarchy.
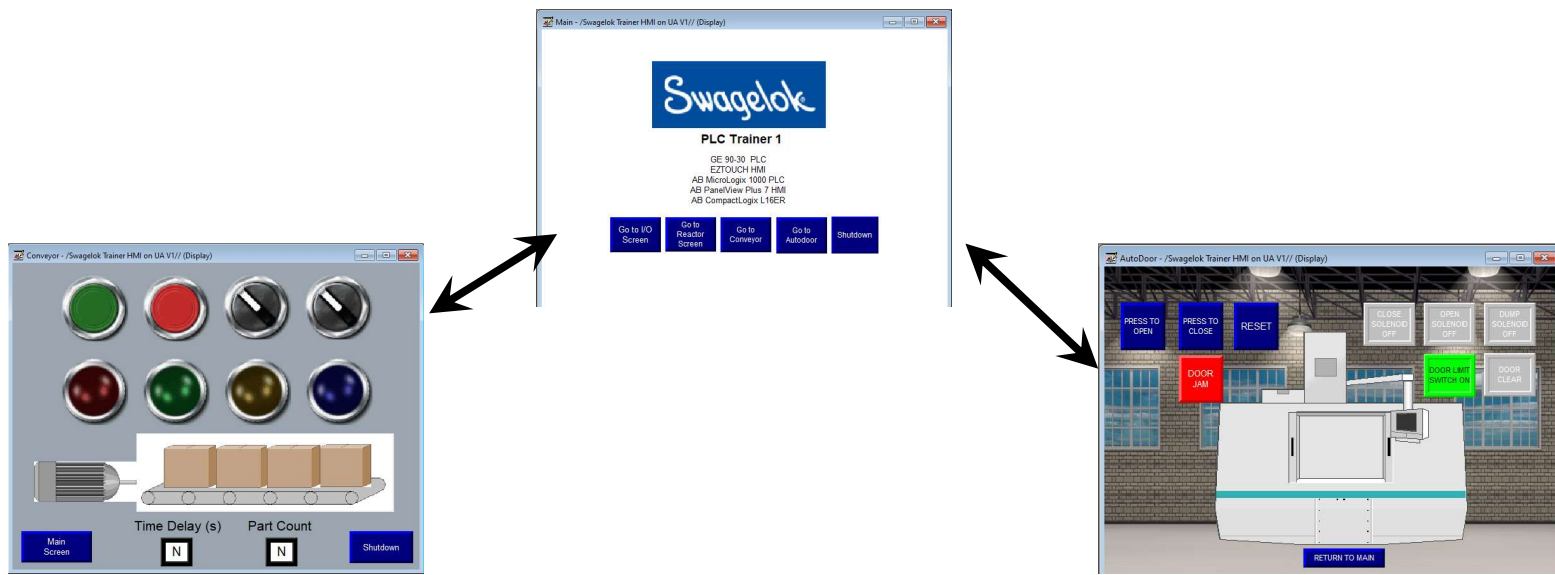
Create a template to ensure consistency

Swagelok

- It is possible to keep a consistent appearance among all the displays in an application by presenting the same pieces of information in the same place on each graphic display.

- To ensure uniformity, develop a display with common elements that acts as a template.

    – Each time you develop a new display, start with a duplicate of the template.
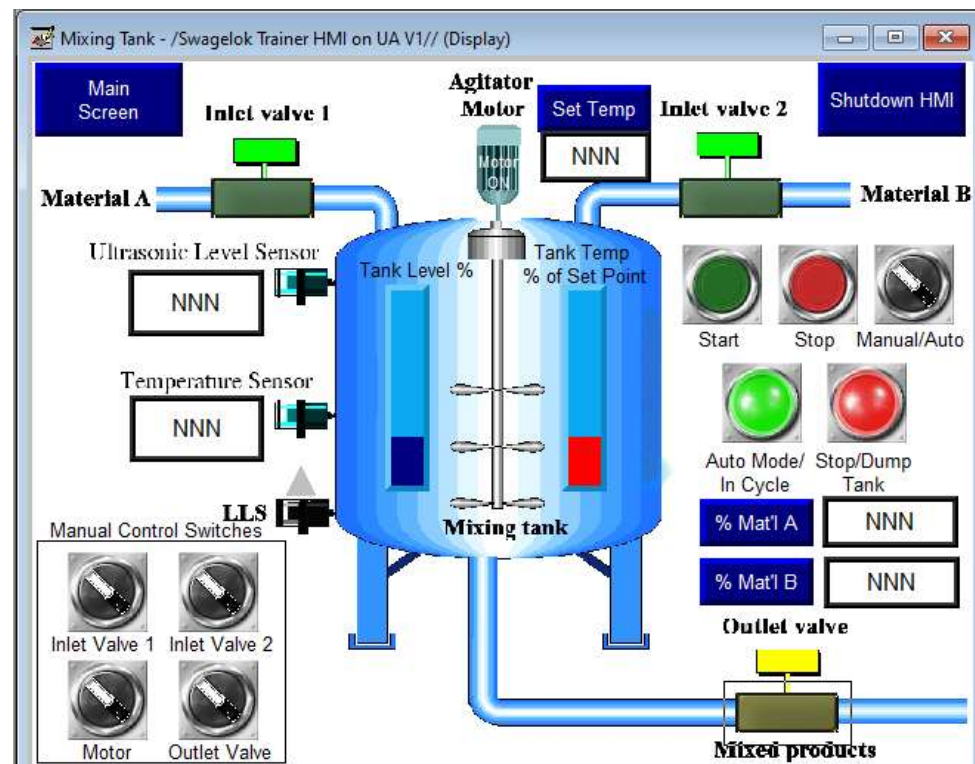
For example, the template could contain:
- Your company logo
- A title
- The date and time
- Navigational buttons

Display Design

Swagelok

- Applying good visual design principles helps users and increases their efficiency.
  - Consistency
  - Clarity
 - Display Types
  - Usability

Consistency

- Be consistent with your use of symbols and color.

- Be consistent with button labels and button placement. When you design several displays, place the same kinds of buttons on the same positions.

  – For example, if you have a **Start** button in a certain position in one display, don't put a **Stop** button in the same position in the next display.

Clarity

- Use easily recognizable symbols. For example, use the conventional ISA symbols for tanks and valves.

- Don't overload the screen with information.

- Use standard, clear terminology, and avoid abbreviations or acronyms that the user might not understand.

- Use colors with recognizable meanings.

  - For example, in North America the colors red and green usually mean stop and start. Keep color meanings consistent by assigning red only to Stop buttons, and green only to Start buttons.

  - Some people are color blind to red and green so don't rely on color alone to establish meaning.

- Use high contrast color combinations, such as yellow on blue.

Display types

- On **Top display type** is used to position the display in front of any other display that is already open at runtime.

    - Can be any size or in any position

- **Replace display type** is used to have the display close any open displays on the screen at runtime.

    - Are the same size as the Project window size set in the Project Settings dialog box.

    - Replace displays do not close displays of the type On Top that have the **Cannot Be Replaced** option selected. Only one Replace display can be open at a time.

Usability
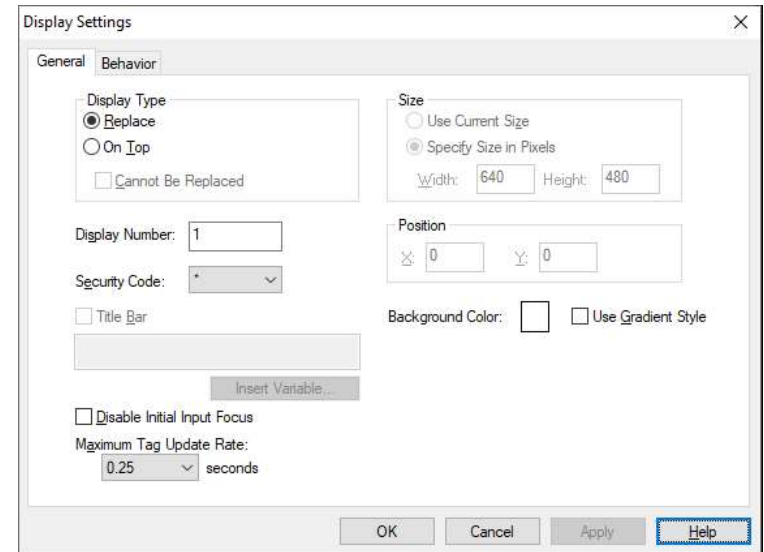
- If you're designing for a touch screen, don't place important buttons where they'll be blocked by an On Top display.

  - The user can't press a covered button. Also, ensure the button is large enough for users to touch easily.

- Ensure there is always a clear way to move between displays.

- Keep the intended user in mind and design displays so they are easy to understand and use.

  - Ask the users to test the displays.

Plan languages

- Which languages you need to use
- Which Windows fonts support these languages
- How different languages will affect design elements in your graphic displays, such as object size and message length
- Whether operators need to switch languages at runtime, and if so, under what conditions.
  - This will help you determine where to locate Language Switch buttons in your application
- How to show operators which button to press to switch to their languages.
  - For example, by using a text label in French or a French flag, to alert a French operator

Plan Alarms

- Before setting up alarms, consider the following:

  - What conditions will trigger alarms

  - How operators will be notified of alarms

  - What information you want alarm messages to contain

  - How operators will respond to alarms

  - The maximum number of alarm messages supported at runtime

  - PanelView Plus 7 Standard applications have a limit of 500 alarms
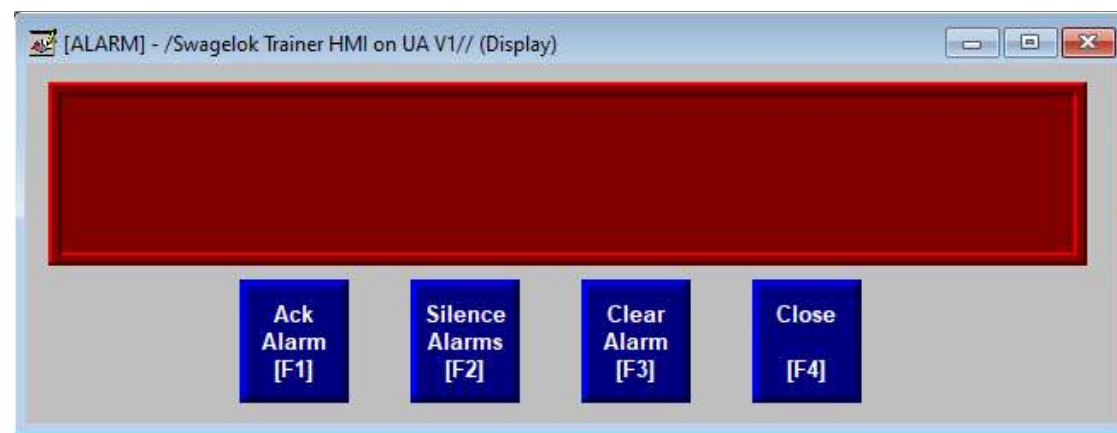
Provide information for the operator

- In addition to notifying the operator of alarm conditions, you can provide information and instructions about plant processes, and inform the operator about system activities.

- Before setting up information notification, plan:

  - What system activity the operator needs to be informed about

  - What conditions will trigger information messages

  - What information you want the messages to contain
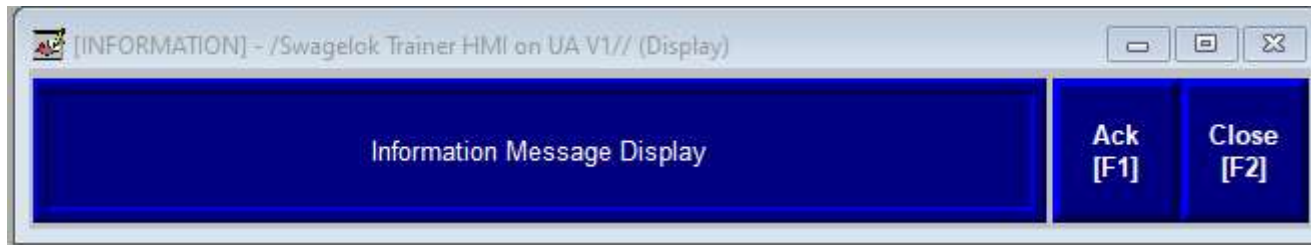
Plan trends

- When planning trends, consider how they will be used. For example, will the trend be used to:

  - Analyze process trends?

  - Monitor production efficiency?

  - Archive process variables to ensure compliance with government regulations?

- Based on such considerations, you can determine:

  - Which tags need to be plotted on the same trend.

  - Which tags need to be logged by using a data log model.

Plan recipes

- When planning recipes, consider how they will be used. For example, will recipes be used to:

  - Compare tag values to pre-set data values?

  - Upload tag values to existing or new data sets?

  - Download values from data sets to tags?

  - Archive tag values to ensure compliance with government regulations?

- Based on such considerations, you can determine:

  - Which tag sets and data sets to pair together in a recipe file.

  - Which RecipePlus button actions to use with the RecipePlus selector and RecipePlus table.

Design a secure system

- When deciding on your security requirements, consider whether to:

  - Restrict access to every graphic display to prevent accidental changes.

  - Restrict access to certain graphic displays.

  - Have everyone log on.

  - Use passwords, and if so, whether to have a minimum length, and whether to require that the passwords are changed periodically.

  - Allow only authorized users to shut down the application.