Section 4: Building an HMI

COGNEX

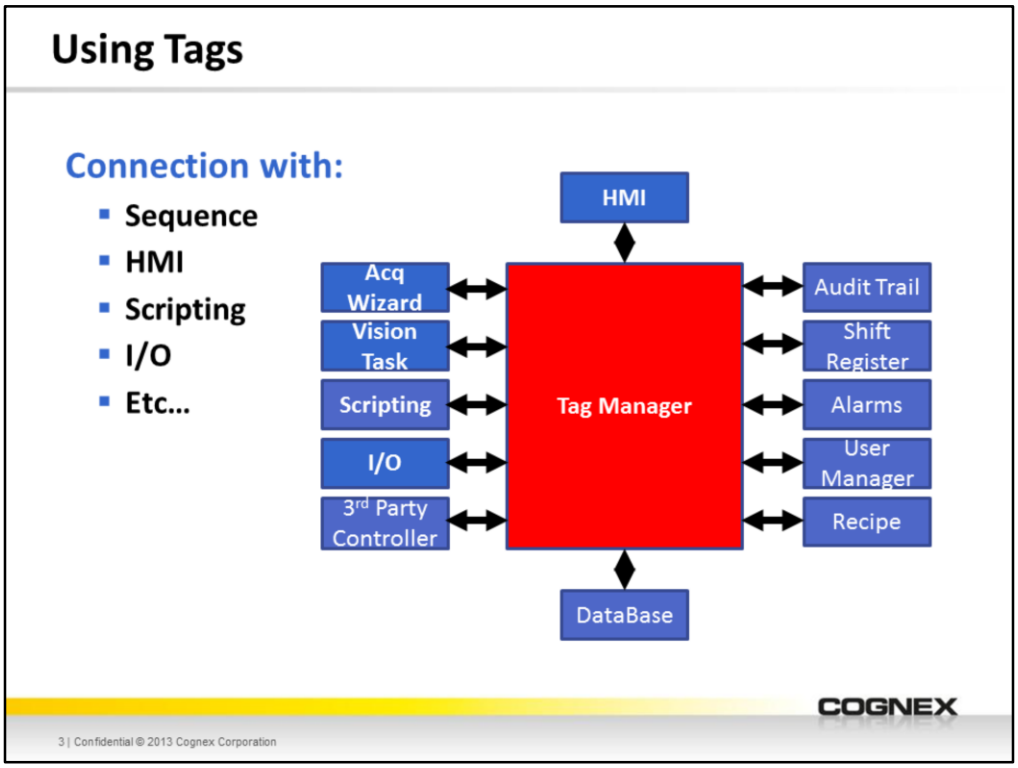
Objective

Building a Human-Machine-Interface (HMI)

- Explore the power of Tags
- Learn about the objects in the HMI interface
- Understand how to display images – both 2D and 3D

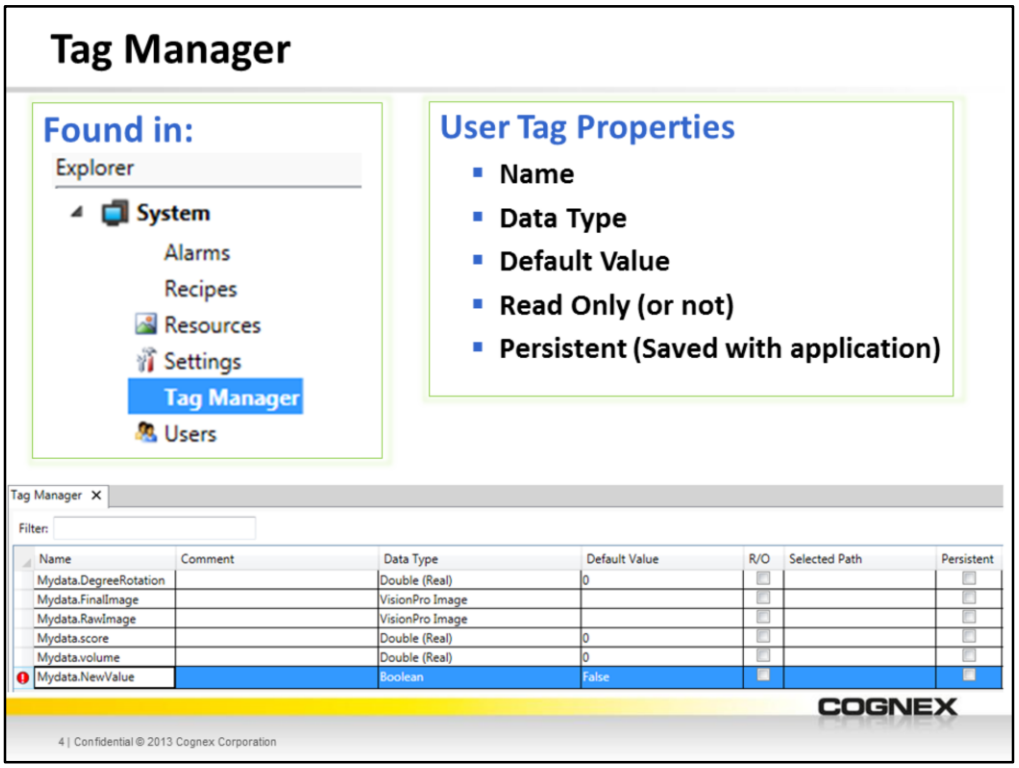
❖ Lab: Create an HMI that displays 2D and 3D images as well as a button to run the Sequence

COGNEX



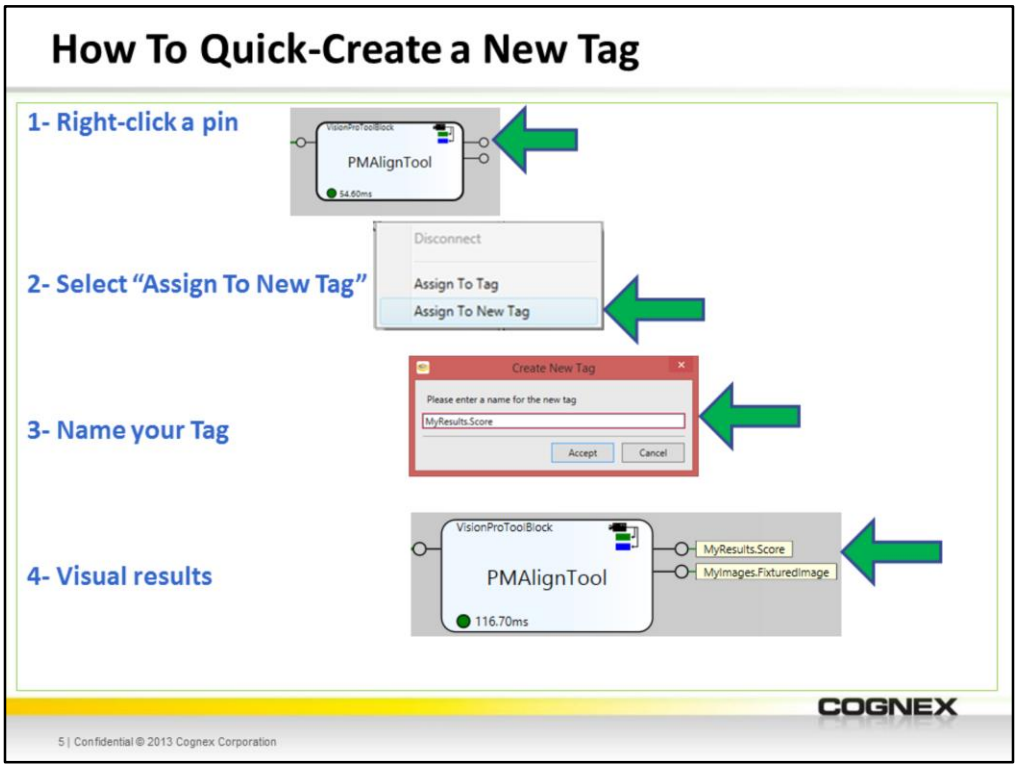
Tags are the “glue” for the application. It allows values to be transferred among objects such as the HMI to the Vision tools. There are many way to create as well as attach to tags.

Tags can be Inputs (read to) or Outputs (written from). It is important to which one is being used.



Tags can be managed through the Tag Manager found in the System area of the Explorer palette. As tags are added to the application, they are also added to the Tag Manager. We can also go here to create a Tag with which we can associate a value later.

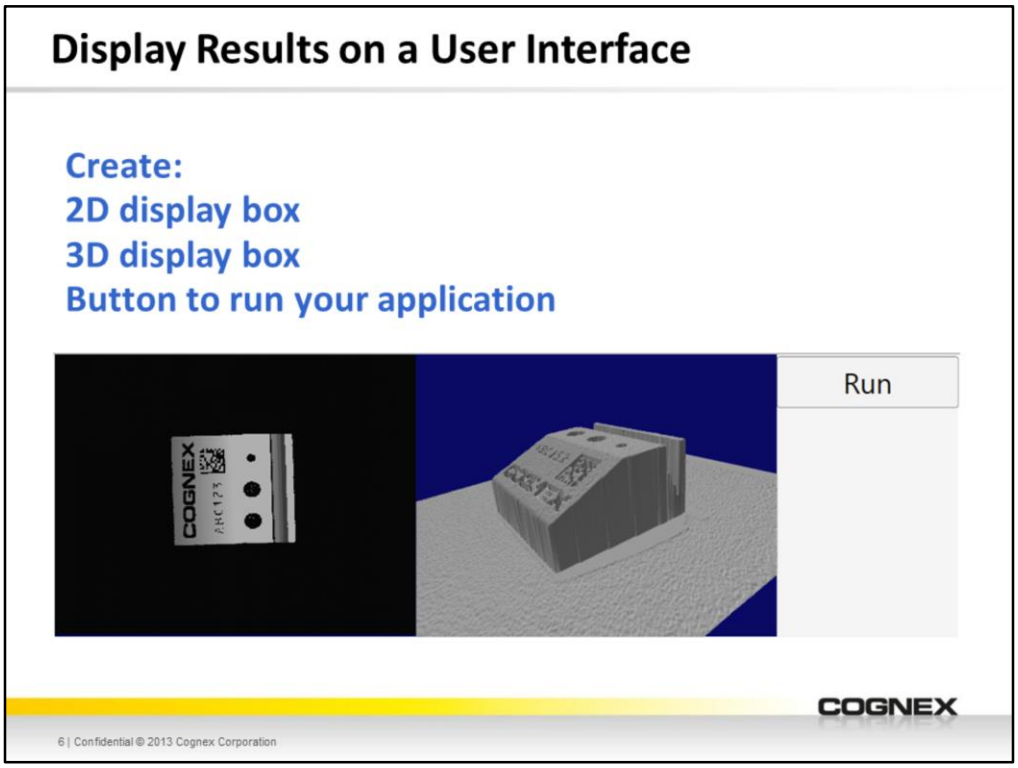
- Selecting R/O will make that a Read Only value so nothing can change it by mistake.
- Selecting Persistent means that the value is saved with the application during a Save command. Otherwise the default value will be saved.
- Selected Path: is used with “Audit Trail” path points to object queried every N ms as set in Audit Trail.



We can create your own tags to allow access to certain results inside our application. Not only does it label the pin on the Sequence, it also adds the Tag to the Tag Selector list so that our value can be used in other programming or scripting areas.

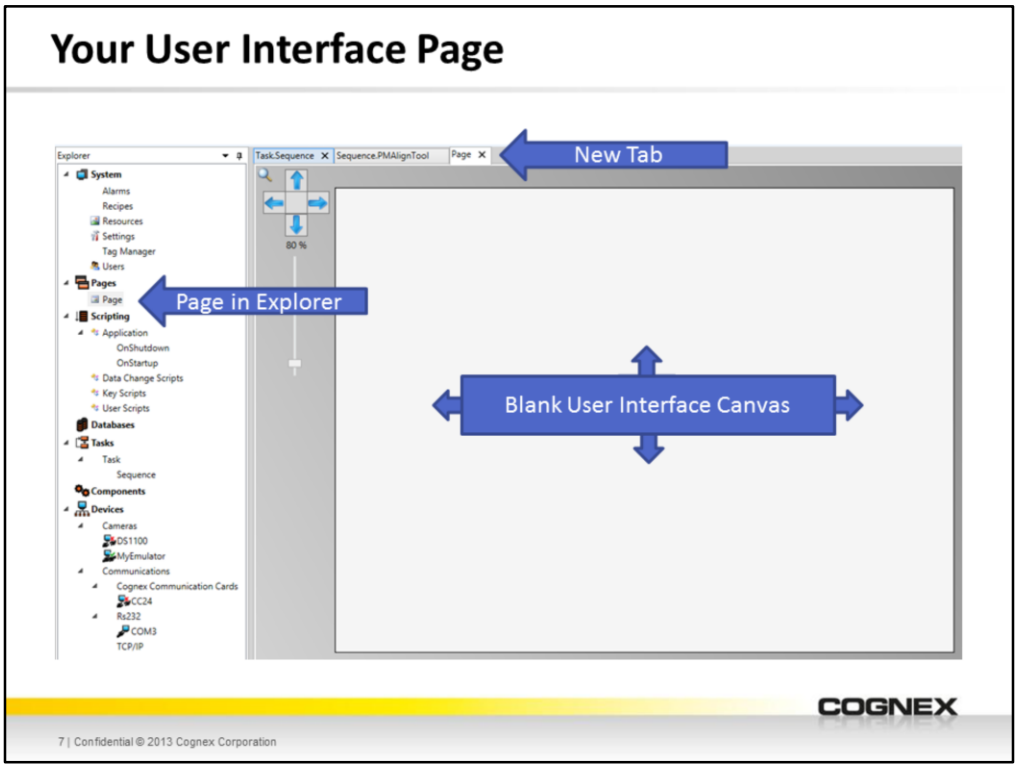
> **Tip:** Saving with a period in the name can assist with finding the variable later. You can break them into groups for faster selection.

In this example, an output tag is being created. It means that the Vision object is writing data to the tag to be used elsewhere.



We are creating an HMI that will allow us to:

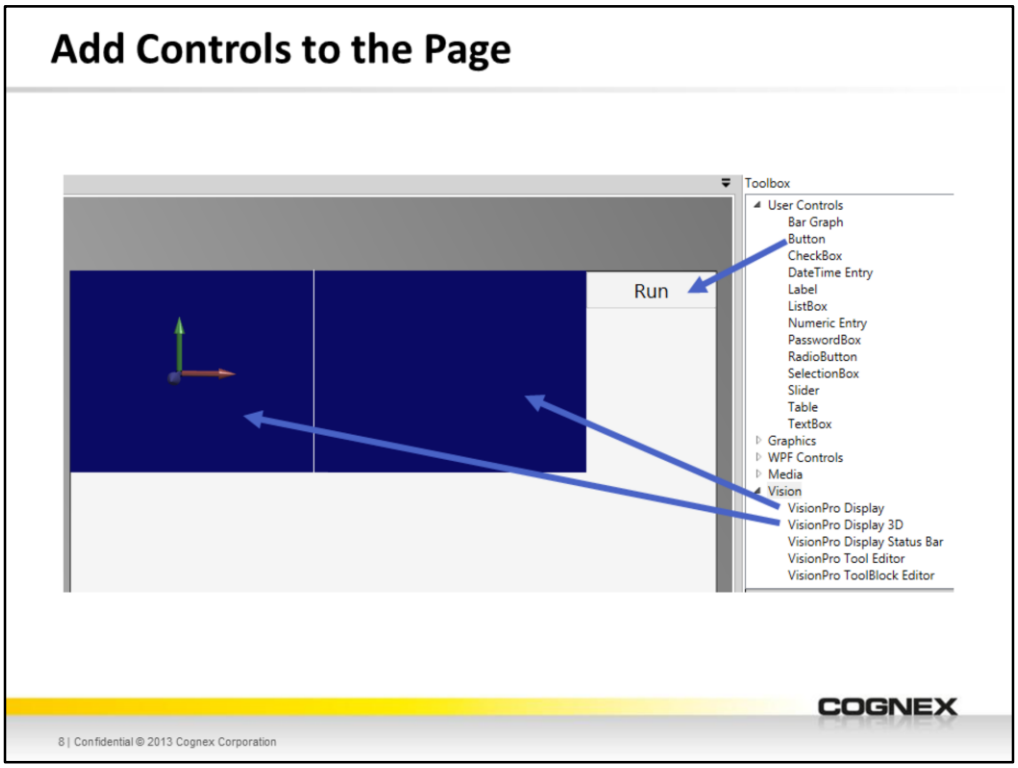
1) Display a 2D image
2) Display a 3D image
3) Use a button to run the sequence



To access the 1st page of your HMI, double-click on the Page icon inside the Pages category of the Explorer listing.

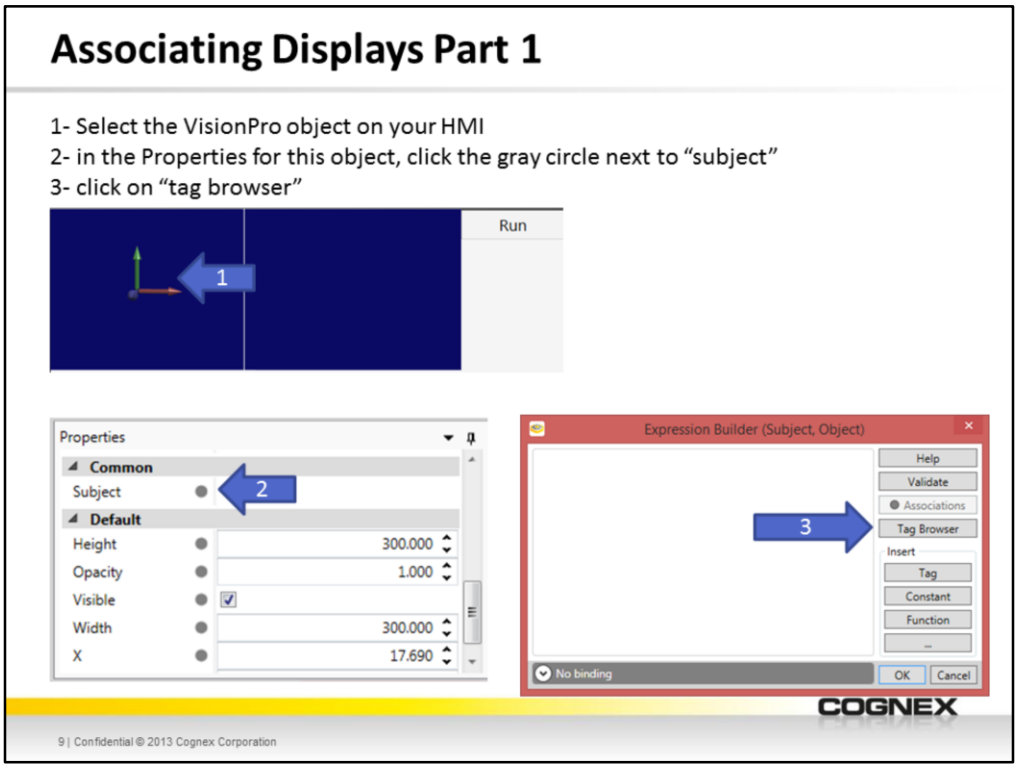
This will display the default page of your project.

This is your blank canvas to design the interface of your application.



Add the following controls to your page:

1) VisionPro Display
2) VisionPro 3D Display
3) Button



Every Vision display has a Subject property.

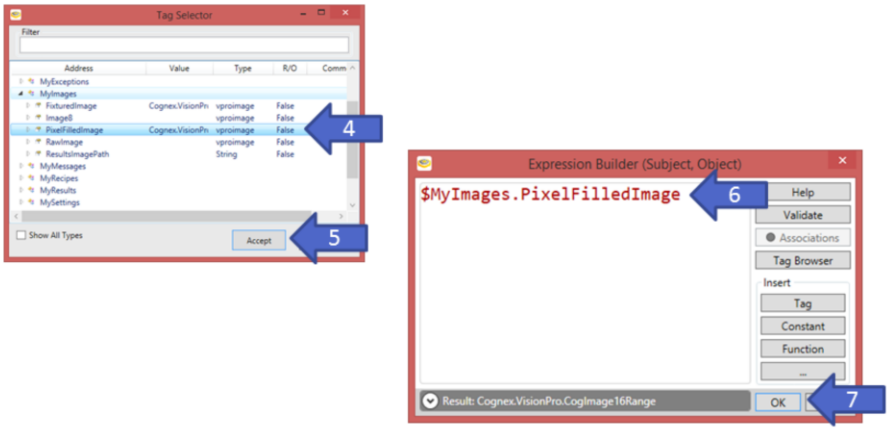
By associating the Subject property with the tag, you can automatically display the desired image in the correct display.

Selecting the dot will bring up the Expression Builder so that the property can be tied to a tag.

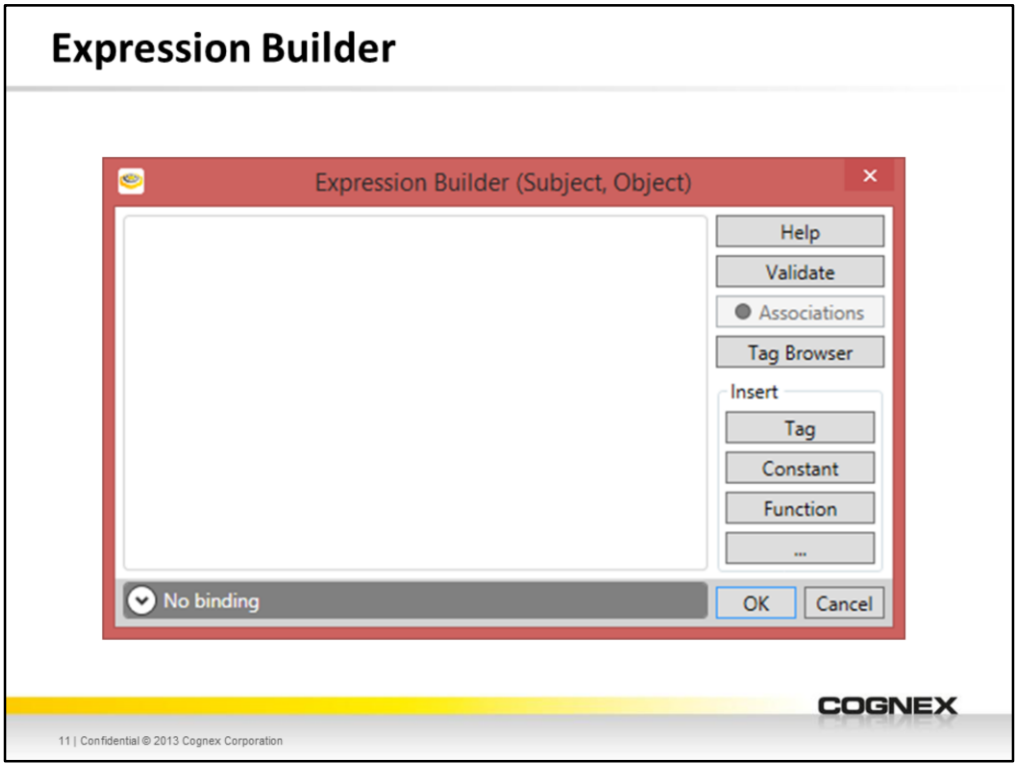
1) Select the VisionPro Display control on the page
2) Find its Subject property and click the gray dot
3) Click the Tag Browser button to select from existing tags

Associating Displays Part 2

4- Browse to, and select, the image tag you have set up previously.
5- Click [Accept] to select the tag
6- See the Tag reference filled into the expression builder
7- Click [OK] to save the tag reference



4) Select the correct tag ($MyImages.PixelFilledImage in this case)
5) Click Accept
6) Verify the correct tag is being used for this Subject property
7) Verify it is of the correct type (CogImage16Range is a 16-bit image)
8) Click OK to complete the setting

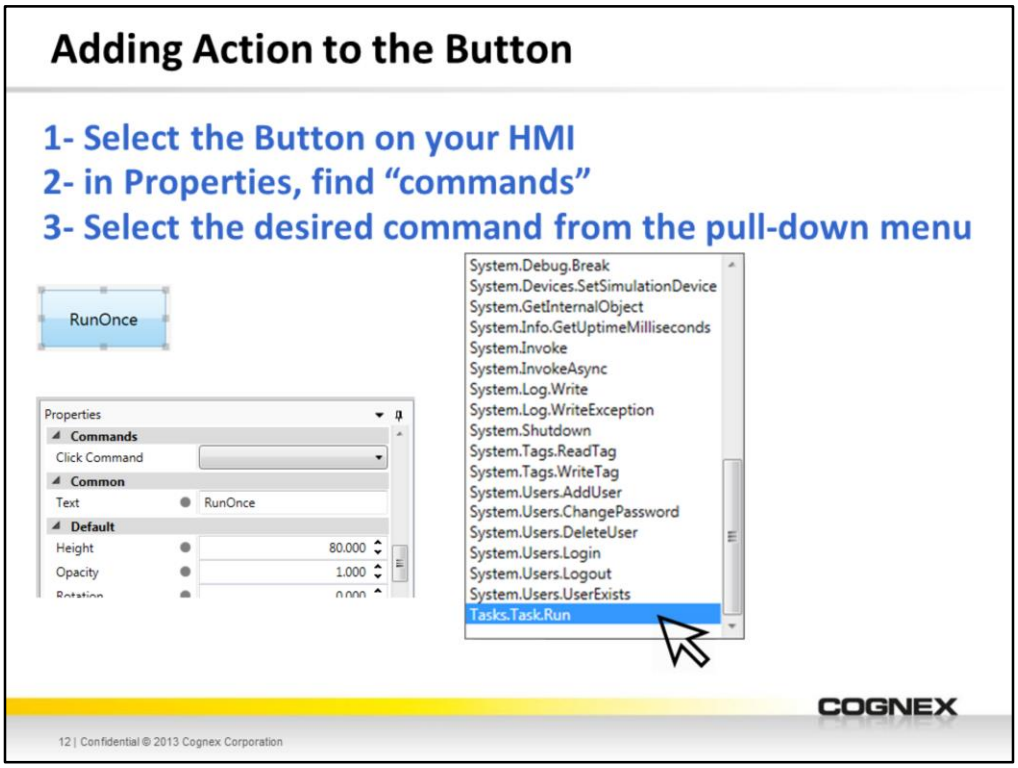


The Expression Builder is a powerful tool that allows you to access different areas and even perform logic statements and custom formatting.

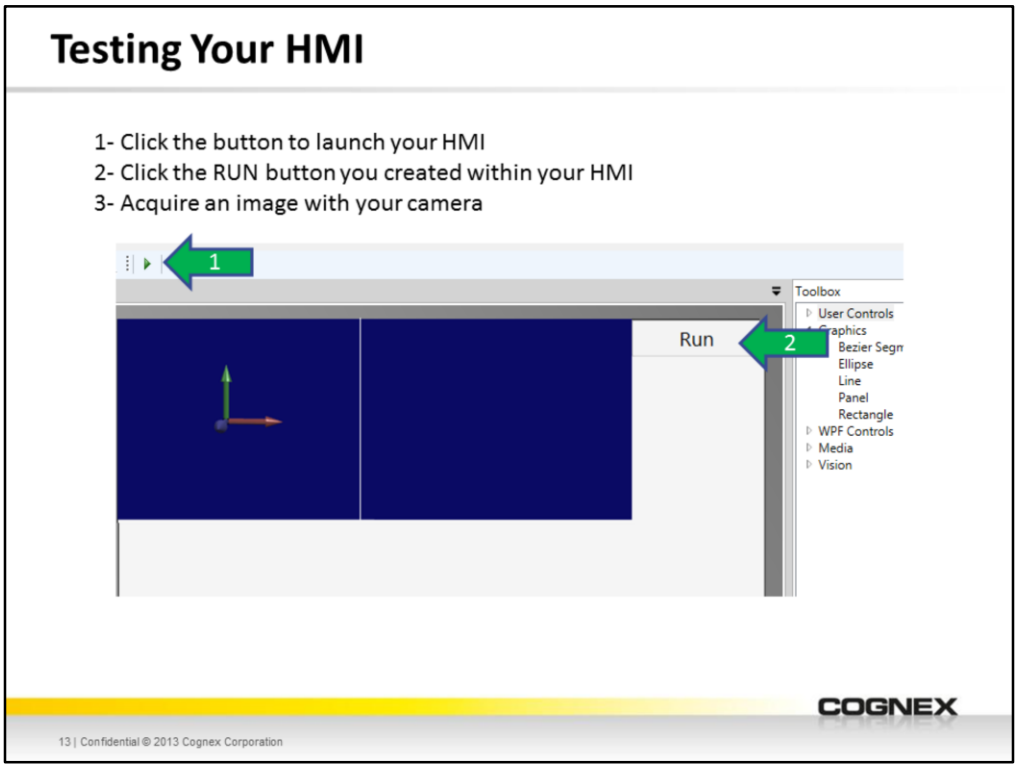
Help: Opens the online documentation to display all the neat things you can do within the Expression Builder.
Validate: Checks your expression for syntax errors.
Associations: Allows expressions to be associated with other values like a background color or a true of false state (more later).
Tag Browser: Allows you to pick from all tags within the project.

Insert Options

- Tag: Quick access to project tags by typing.
- Constant: Access to mathematical constants like pi and exponent.
- Function: Access to logical statements like If.
- …: Access to everything without a predefined category.



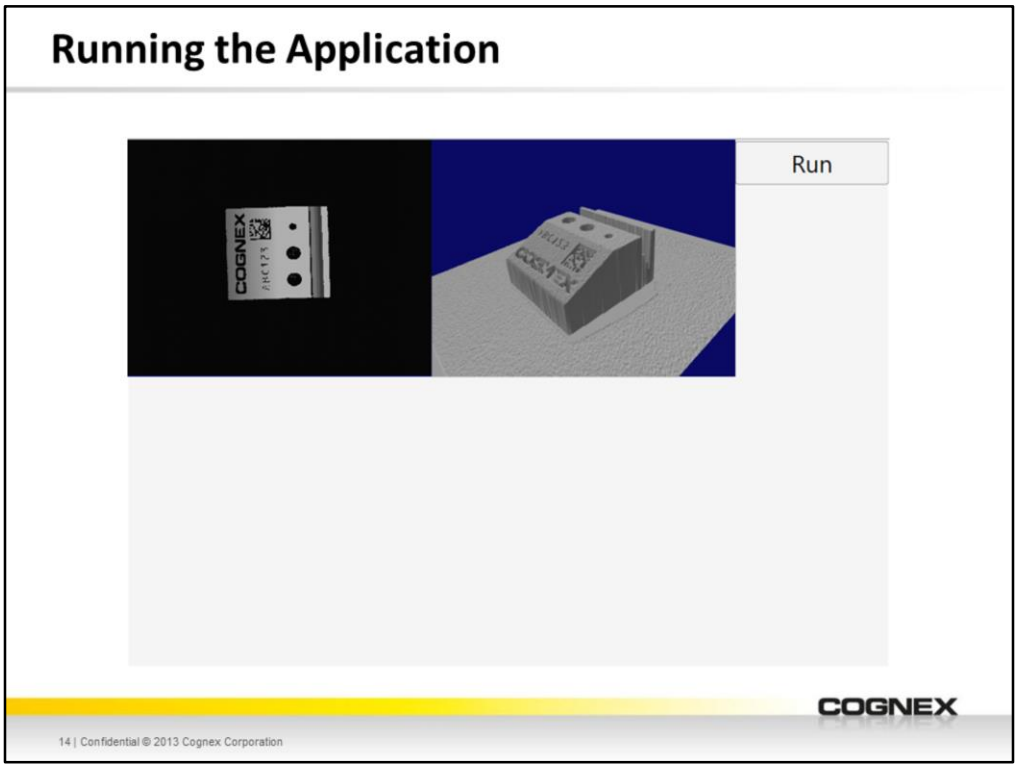
To make a button perform and action or trigger an event, you must set it's Click Command property.
To make this Acquire button run the task that acquires the image, runs the tools, and displays the results, we use the Tasks.Task.Run Click Command.



To test the page where the displays were added, start test mode using the Play button, and go full screen by pressing the F11 key.

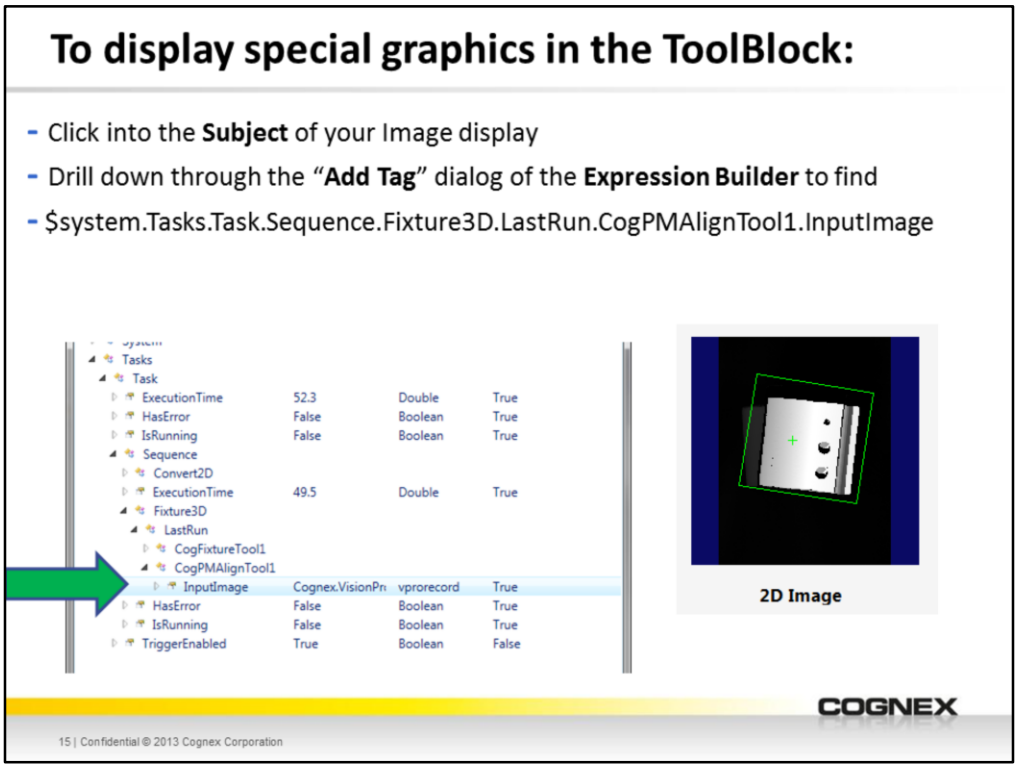
> Note: When you are in Test Mode, the HMI is now active though the rest of the program is "blocked" – you cannot make any changes and running the sequence does not work unless you are running it through controls in the HMI.

Then you can press the "Run" button to acquire a new image. If you are using a live DS1000 sensor, make sure you move it after to press "Run" so it can acquire an image.



Note that the 2D and 3D image are being displayed. We can use the mouse controls as we did in the Wizard to manipulate the images – especially the 3D.

Pressing the “Run Once” button causes the Sequence to run once which calls for another acquisition.



To display the graphics in the ToolBlock, you must drill down through the "Add Tag" dialog of the Expression Builder to select the output image (or rather the InputImage – it's a VPro thing; LastRun.InputImage) of the tool in the ToolBlock to get the graphics to show.

It would be Tasks -> Task -> Sequence -> {Name of your ToolBlock } -> LastRun -> {Name of the tool} -> InputImage

Summary

Building an HMI

- Learned about the power of Tags
- Implemented interactive objects in the HMI
- Implemented both 2D and 3D image displays

COGNEX