

Cognex Designer Advanced – Scripts



COGNEX

Objective

Cognex Designer Advanced – Scripts

- Understand the different scripting points
- Explore implementing different scripts
- Learn how VisionPro handles scripting

❖ Lab: Create a VisionPro tool to convert radians to degrees

COGNEX

Scripts



3 | © 2014 Cognex Corporation

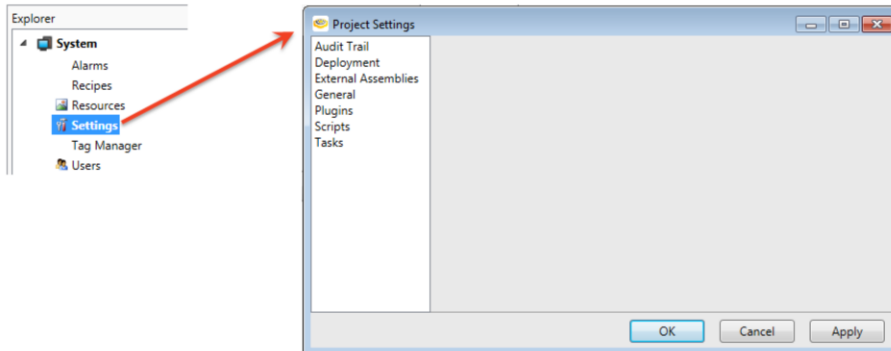
COGNEX

Scripting is used throughout the Cognex Designer application. C# must be used when scripting in Cognex Designer though VisionPro will allow for both C# and VB (within a ToolBlock)

Settings – Global Referencing

Project Setting:

- Track Changes
- Define directory paths and project dlls
- Enable / disable plug-ins



- Audit Trail – Audit Database and table settings. Used to monitor environment and application changes
- Deployment – Deployment folder, scaling options, windows mode
- External Assemblies – add project level custom DLLs / references
- General – Project password, version number control
- Plugins – shows which controls are available and being used. Is currently read only.
- Scripts – Add other DLLs to be used at the project level as opposed to just the individual script level
- Tasks – Garbage collection (GC) settings.

The assemblies in the Scripts tab will be automatically referenced in each script in the project.

However, for convenience, a script added to the Scripts tab is automatically added to the External Assemblies Tab as well, because it is very likely, that if you use an assembly/3rd party type from an external assembly in a script, you will want to create a tags / script arguments of that type as well.

Therefore, when adding a Script reference, we automatically add a global / project level reference to that assembly, so that it can be used in the entire project, where types can be selected.

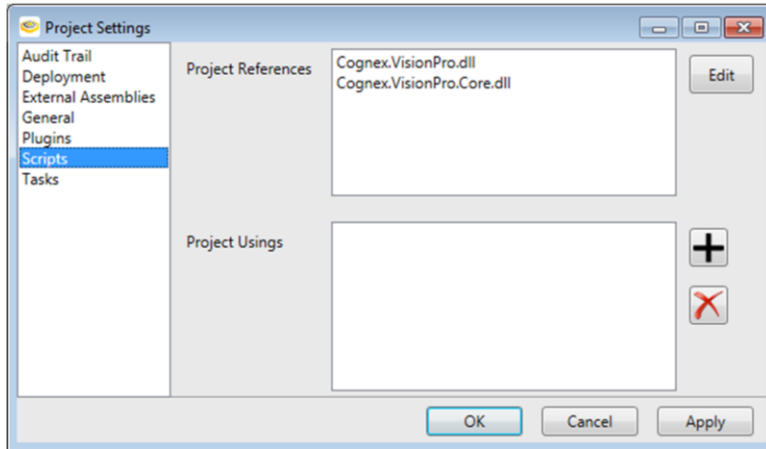
This is not the case however in the other way, i.e. when you add an External Assembly reference, it won't be automatically added to the Script references,

because it is a best practice in programming to keep the references to a C# project (script in our case) minimal.

However after you referenced an assembly in the External Assemblies, it's just a few clicks to add it to the Scripts as well, as external referenced DLLs are listed in the Discovered References (list of "known" DLLs), so the user doesn't need to browse it again:

Global Scope

**All Scripts will have access to these
DLL and Using References**

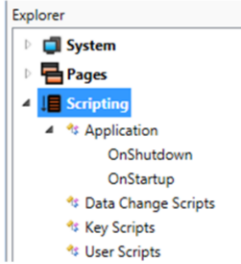


By declaring the library References and the Using References in the Settings area, all scripts will include them as well.

Helpful when many similar scripts use the same DLLs, no need to call them within each individual script.

Press Edit to add additional DLLs than the ones listed.

Add Functionality via Scripting points



- **Also:**
 - Script blocks
 - HMI mouse interaction events
 - On Page show/hide
 - On Task error / completion
 - On Sequence error/completion
 - And many more!

COGNEX

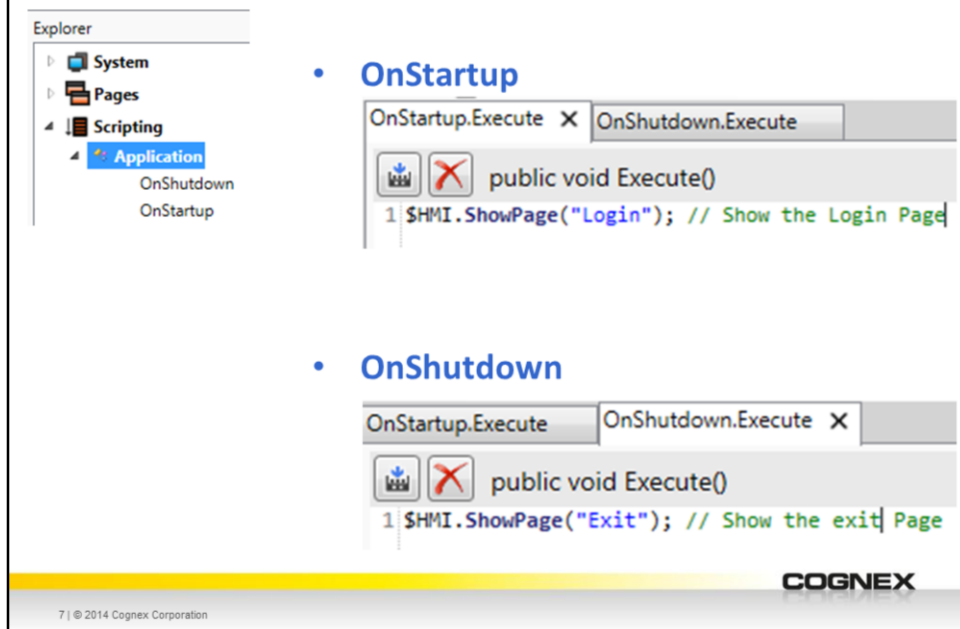
6 | © 2014 Cognex Corporation

Cognex Designer and VisionPro are written in .NET. Because of this, functionality can be added simply through adding code.

The program has certain points at which scripts can be added such as the start-up or shutdown of the application. It can also be during a function call.

This provides a lot of flexibility with control and design of the application.

Application Startup / Shutdown



The screenshot displays the Cognex application development environment. On the left, the 'Explorer' pane shows a tree structure with 'System', 'Pages', 'Scripting', and 'Application'. Under 'Application', 'OnShutdown' and 'OnStartup' are listed. The main area shows two code editors. The top editor, titled 'OnStartup.Execute', contains the following code:

```
public void Execute()  
1 $HMI.ShowPage("Login"); // Show the Login Page
```

The bottom editor, titled 'OnShutdown.Execute', contains the following code:

```
public void Execute()  
1 $HMI.ShowPage("Exit"); // Show the exit Page
```

The Cognex logo is visible in the bottom right corner of the interface.

Using scripting for Startup and Shutdown gives points to control the flow of the application.

- OnStartup can be used to set the correct page of the HMI so the user always sees a consistent start point.
- OnShutdown could be used to save select values or show an end page that may let the user know the application is closing.

Key Scripts

Executed when a Key Combination is Pressed

The screenshot illustrates the workflow for creating and editing a Key Script in the Cognex software. It is divided into two panels. The top panel shows the 'Explorer' tree on the left with the 'Key Scripts' folder selected, and a 'Select Keys' dialog box on the right. The dialog box has checkboxes for 'Control' (checked), 'Alt' (checked), and 'Shift' (unchecked), and a 'Key:' text field containing the letter 'T'. An 'Accept' button is at the bottom. A red arrow points from the 'Key Scripts' folder to the dialog box. The bottom panel shows the 'Explorer' tree with the 'Alt_Control_T' script selected, and a context menu with 'Edit', 'Export', and 'Delete' options. A red arrow points from the 'Edit' option to a script editor window titled 'Alt_Control_T.Execute'. The editor shows a code snippet:

```
public void Execute()  
1 $System.Shutdown();
```

COGNEX

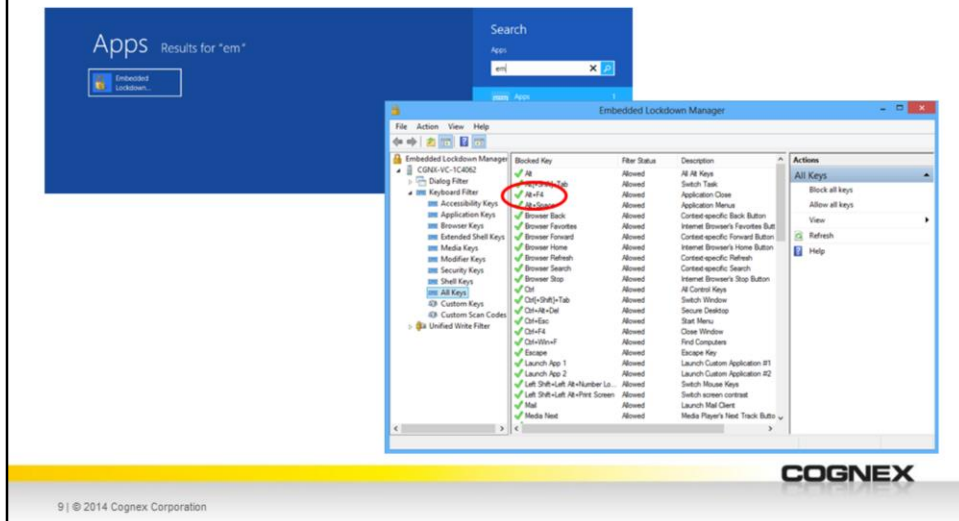
8 | © 2014 Cognex Corporation

Key Scripts can be used to add functionality when a certain combination of keys are pressed.

This can be advantageous in the application to allow the user to do things such as force an acquisition, save specific changes, or exit from the application.

Embedded Programs

Preventing Alt + F4 from Shutting Down

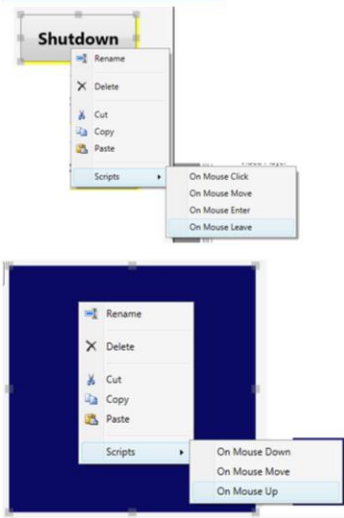


The Controller runs Embedded Windows 8 as its operation system. Though it is not a full installation of the operating system, it does have some default embedded programs such as keys performing certain functions. One of these keys is ALT+F4 which will shutdown the current program.

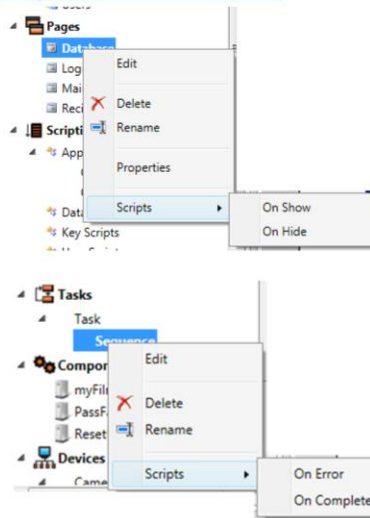
There is a way to disable this on the VC-5 Controller is to select the Windows key -> start typing "emb" -> select the Embedded Lockdown Manager application. You can now choose which embedded options to disable.

Right-click exposes that many Objects have Script points

HMI elements



Explorer elements



COGNEX

10 | © 2014 Cognex Corporation

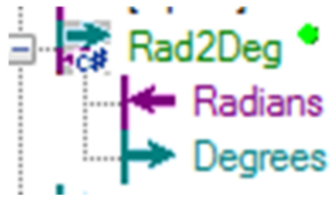
We have seen how HMI objects like a button or a display has scripting points.

Items in the Explorer have scripting points as well. Code can be set to run when a page opens or closes as well as if there is a error on the Sequence.

To access these scripting points, right-click on the objects and select the Scripts selection.

Scripting in VPro

Rad to Deg



$$(\text{Radians} * 180) / \pi = \text{Degrees}$$

COGNEX

11 | © 2014 Cognex Corporation

Every result angle from VisionPro is outputted as a radian value. Even if “degrees” was selected in the form, this is only a conversion for the form and the actual value is in radian.

It would be nice to have a way to quickly convert radians to degrees.

Reasons for Scripting

Extend the Capabilities of VisionPro:

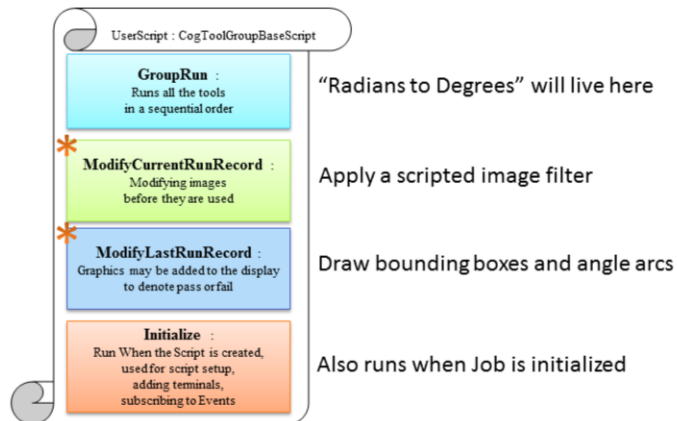
- Create a “Custom Tool” to perform an action not available
 - Add two numbers
 - Perform mathematical, logical, or string manipulation
- Stitch acquired images together
- Conditionally run tools or change the execution order
- Situations in which you need to do the same thing many times (avoid Jobs with hundreds of tools and terminals)
- Change or add the Graphics related to a particular inspection

COGNEX

12 | © 2014 Cognex Corporation

- Create a “Custom Tool” to perform an action that there is no VisionPro tool for.
 - Add two numbers
 - Perform mathematical, logical, or string manipulation
- Expose parts of the API that are not available via Tool Terminals.
 - CogBlobResult.BoundingBox()
- Easily allow an AppWizard application to have additional control over your inspection routine
 - Add a flag to retrain a PMAlignPatten
 - Switch between different modes in the same job
- Stitch acquired images together
- Conditionally run tools or change the execution order
- Situations in which you need to do the same thing many times (avoid Jobs with hundreds of tools and terminals)
 - MultiTarget Sample
- Evaluate a complicated RunStatus expression of an inspection much easier than using the ResultsAnalysisTool
- Save particular images to file
- Write to a log File
- Change or add the Graphics related to a particular inspection
- Creative uses to solve problems the designers of VisionPro never anticipated

ToolGroup Advanced Scripting Access Points



* - Available in advanced scripting

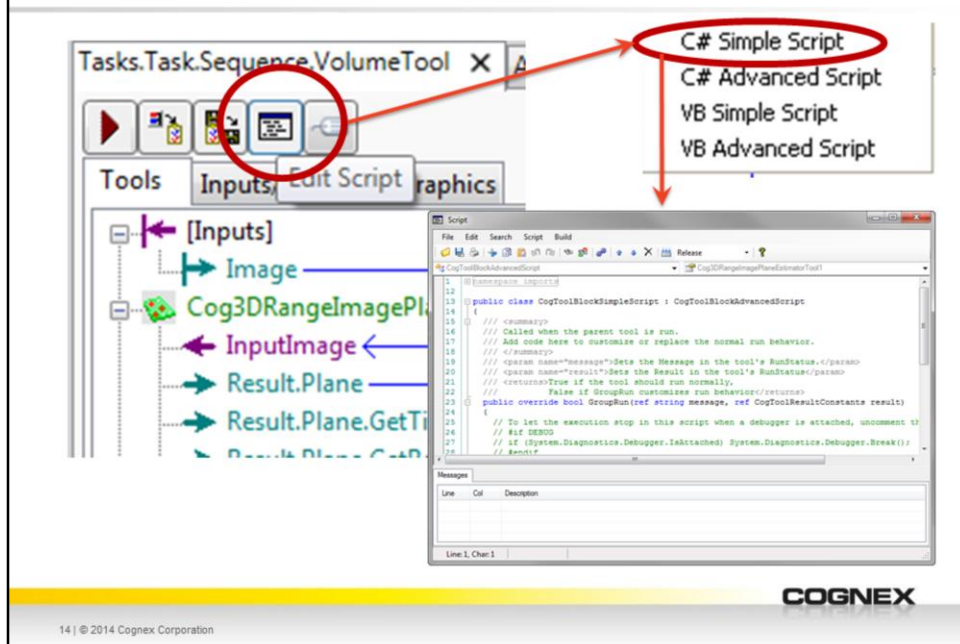
COGNEX

13 | © 2014 Cognex Corporation

Scripting in the ToolBlock gives 4 different points of entry:

1. **GroupRun** – this is the core of the ToolBlock. By default, it runs all the tools in a sequential order.
2. **ModifyCurrentRunRecord** – this is the current record created by the ToolBlock and tools as they start to process.
Modifying images before they are used would happen here.
3. **ModifyLastRunRecord** – is the result record created by the ToolBlock and the tools.
This would be where graphics may be added to the display to denote pass or fail
4. **Initialize** – this creates the terminals that are used to bring data in and get data out of the toolblock

Opening Script Editor for Tool Group



Pressing the Edit Script will gain you access to the script area for the ToolBlock.

If you are accessing it from the ToolBlock Object (upper level), only C# is available.

If you create your own ToolBlock as a tool within the ToolBlock Object, then both VB and C# are available

Simple Scripting is used to create custom tools like logic or data manipulation. It is very easy to add functionality this way.

Advanced Scripting gives access to all the different Access Points. It gives more flexibility

Creating Your Own Tool

Create Inputs and Outputs

Inputs

Outputs

Name	Type	Value
Radians	System.Double	1.54976

Name	Type	Value
Degrees	System.Double	88.795

2.3605ms 2.4571ms

COGNEX

15 | © 2014 Cognex Corporation

Add the input and output data points that you will need to use within the script as variables

The Inputs/Outputs tab allows you to create:

- System variables

 - Integers

 - Doubles

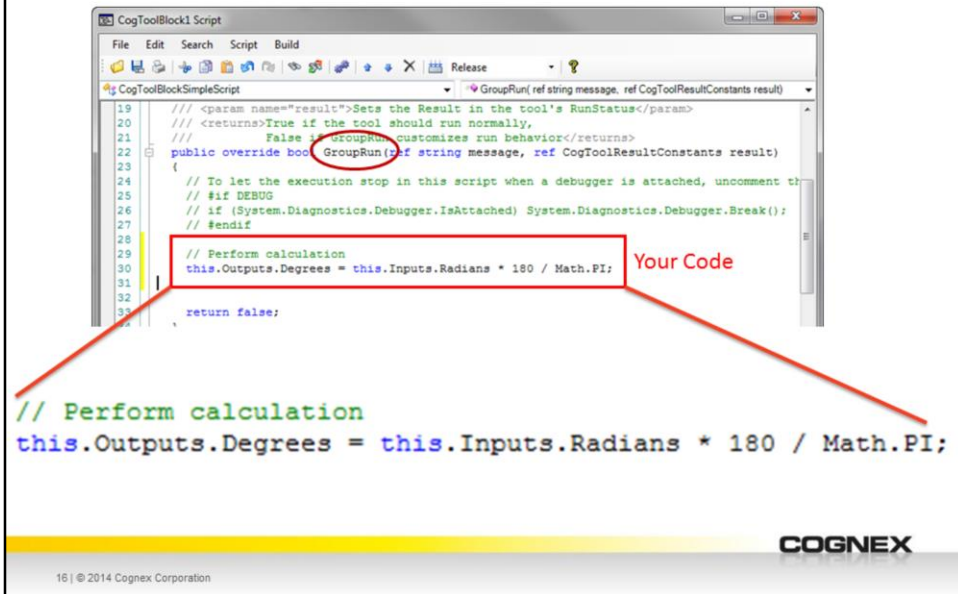
 - Booleans

- VisionPro variables

 - CogImage8Grey

 - CogToolResultConstants

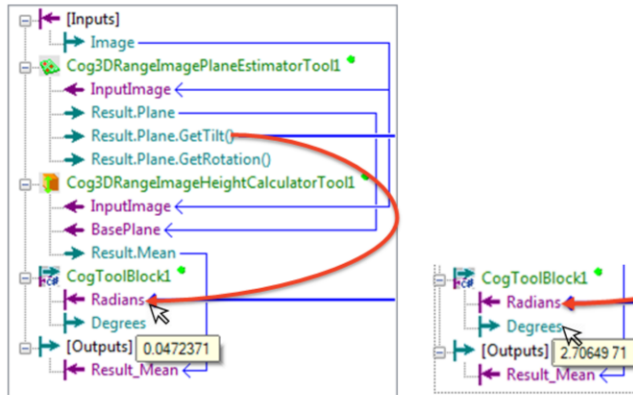
Add the "Guts"



All we are doing is taking the Radians input and calculating the Degrees equivalent.

```
// Perform calculation
this.Outputs.Degrees = this.Inputs.Radians * 180 / Math.PI;
```

Attach an input and Check



COGNEX

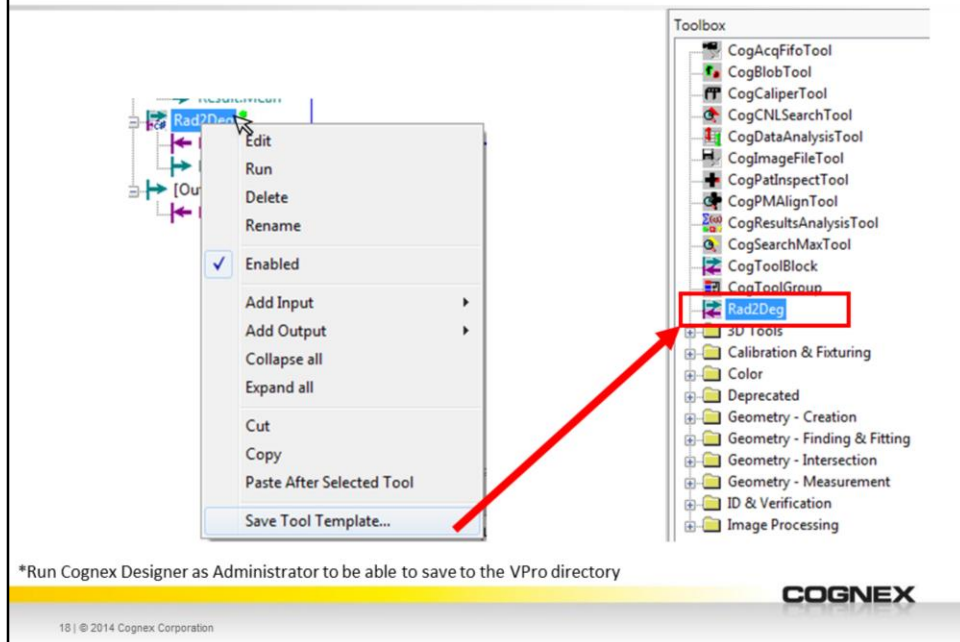
17 | © 2014 Cognex Corporation

We are taking the angle input coming back from the Tilt of the PlaneEstimator tool and converting it to radians.

Note the flyover graphics that show the input of radians (0.0472371) and the output of degrees (2.70649).

This can be attached to the [Outputs] of the ToolBlock Object so that it can be displayed on the HMI.

Rename the Tool and save to ToolBox for future use



Right-Clicking on the ToolBlock will bring up the option to save the ToolBlock as a Tool Template. The tool will now reside in the ToolBox for VisionPro and be available for subsequent applications.

You **MUST** run Cognex Designer as Administrator to be able to save to the VPro directory.

Note: You can remove old templates by deleting the appropriate file from the ..\VisionPro\bin\Templates\Tools directory.

Summary

Scripts

- Explored different scripting points
- Implemented a script
- Discussed how VisionPro handles scripting

COGNEX