The DockablePluginPythonQt for EL Console

User's Manual: Software

Rev.1.0.2 May. 2016

TABLE OF CONTENTS

LIST OF FIGURES

1. Overview

1.1 Features

This document describes how to setup and use the Plugin (DockablePluginPythonQt_API6.dll).
We create three demos for the Plugin, they will help you how to use the basic controls, change axis value for 1D/2D Table and change cell value for 1D/2D Table. We are using Visual Studio 2013, XML, Python and Qt 5.4.0 to develop the Plugin.
For the first requirement, the XML format will store all information relating UI in one file, each model only has one XML resource script file. By modifying code of this file and reload the Plugin, you can change your Model UI at run-time without shutdown the EL Console application.
For the second requirement, read data from XML resource file, using the Qt to create the controls on main window.
For the third requirement, the Plugin supports two ways for setting or getting value of channels, using XML script and Python script.

- With the XML script way, you only need to define information of channel variable type (not all variable types are supported) in resource file and the Plugin will do the rest for you. This approaching is really useful in case you only need to set/get value for your channel. However in case your channel variable type is not supported or you want to do anything else, further than set/get action? Python script is your choice!
- With the Python script way, the Python script will run when there is an event raised from Win32 Control of your Model UI. We also support some exported functions from the Plugin, so you can use them to interact with your model through ctypes module in Python or you can add more code for your purpose.

1.2 Scope

This project has three demos, the developer can custom the Plugin for their purposes.
Following the task requirement, the Plugin is designed with two purposes:

- Allow developer to create a customizable UI.
- Allow developer to handle their task, they can modify XML resource file and Python code.
- Allow developer to debug through a debugging dialog.

1.3 Environmental Requirement

Table 1.1 Environment Requirement

Equipment	Explanation
Windows Host PC	Windows 7 (32bit or 64bit) Windows 8 (32bit or 64bit)
Python	Python-2.7.8
EL Console	EL Console V97.8427
Installer	Installer for the Plugin

2. Plugin block diagram

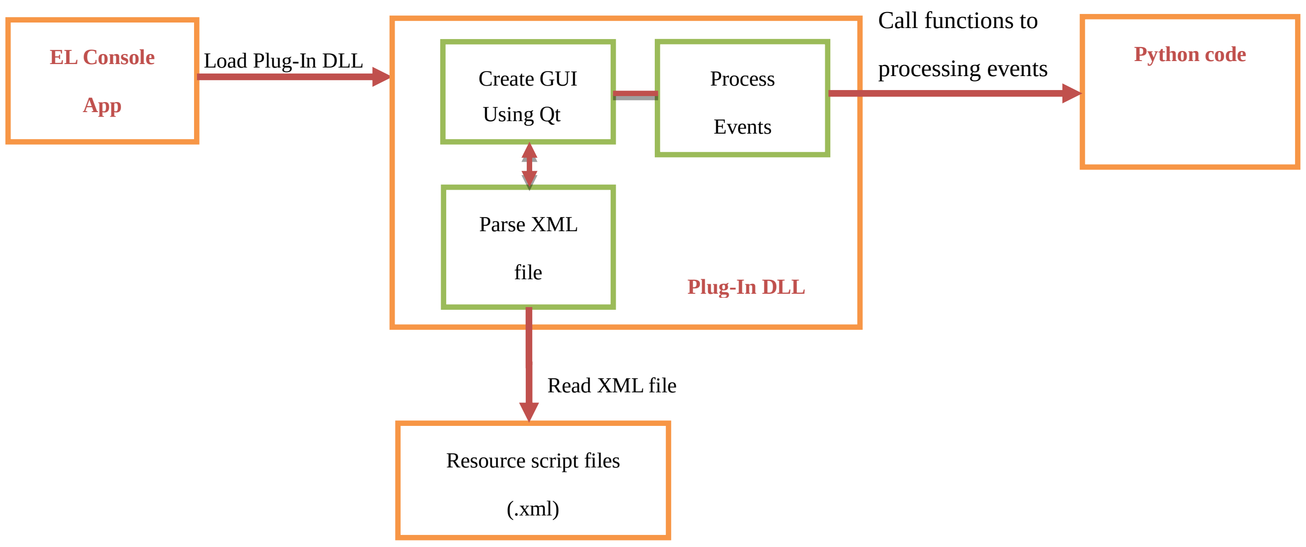


Figure 1-Plugin Block Diagram

- **Resource script file (.xml):**
  a) Store all information relating UI.
- **Plug in dll:**
  a) Use QtXmlParser to parse XML file, resource data will be stored in data structure.
  b) From resource data, using QWinWidget and Qwidget to create a main window for custom interface.
  c) Hook the events were defined in XML to the control, and call the event functions in the python code.
  d) Re direct debug information to debugging dialog.
- **Python code:**
  a) Program behavior of Model.
  b) Support the functions to handle the developer task.

3. Set up the Plugin for testing

3.1 Install EL Console version 97.8427

Make sure that you install right the EL console **version 1.97.8427**, this version supports API el_dockable_dll. To install the EL Console version version 1.97.8427, you can find setup.exe file on the SVN at the location:
…\el-public-qa\trunk\Apps\Builds\EL Console\V97_Version6API
After install EL Console app and launch the app, the screen will be displayed as below.

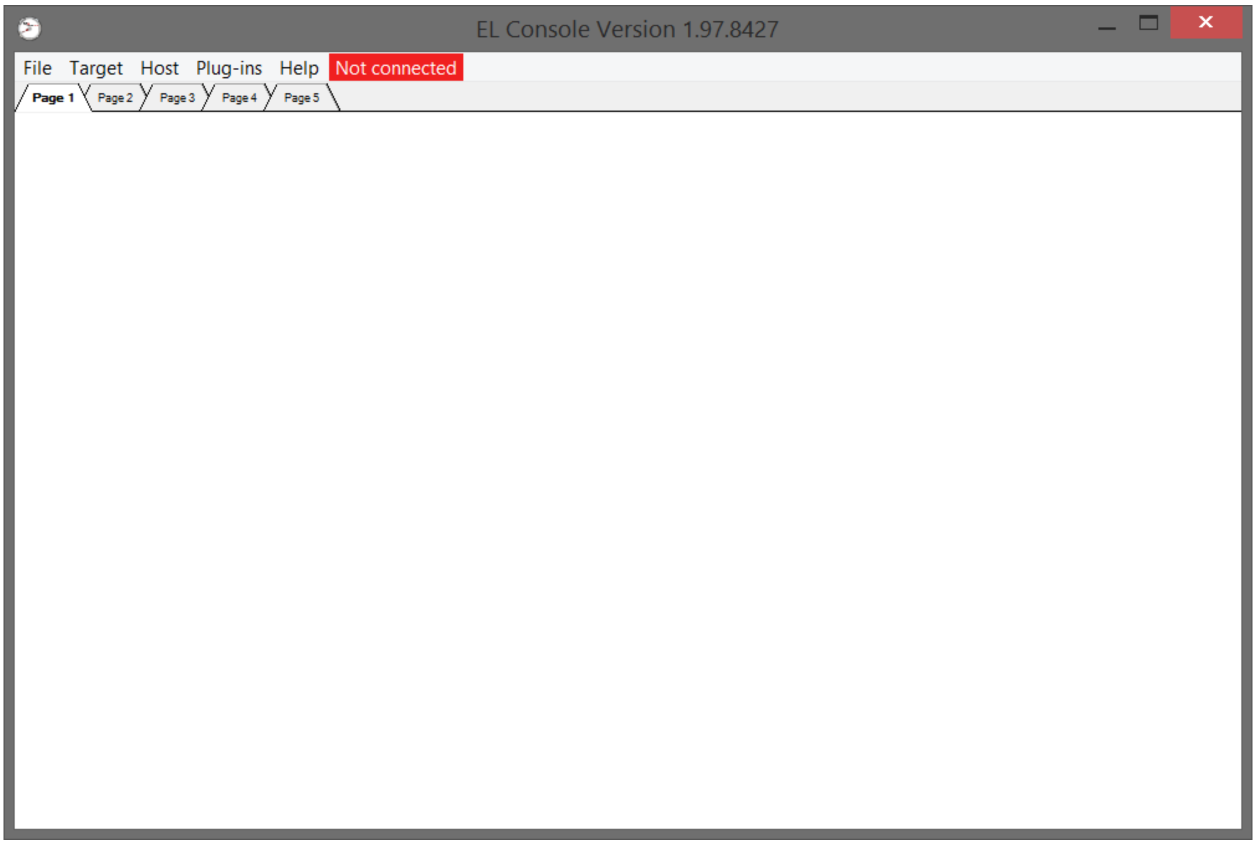


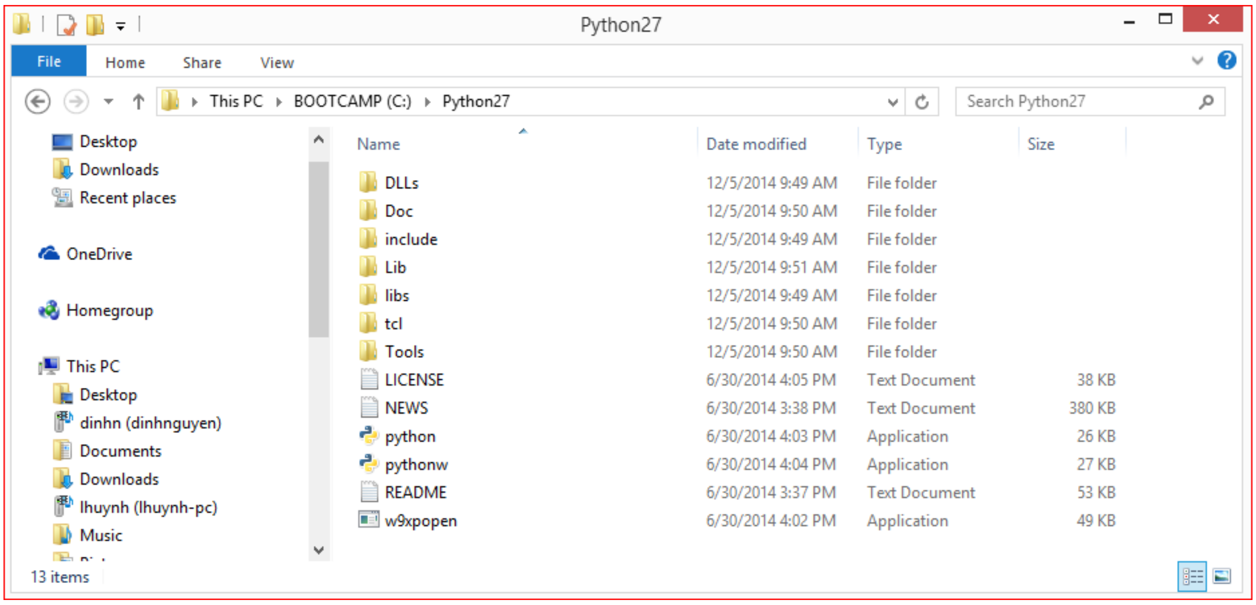
Figure 2-The screen of the EL Console

3.2 Install the python-2.7.8

To install python-2.7.8, you can find file (python-2.7.8 .msi) in the Python-2.7.8 folder on the SVN at the location:

...\el-external-shc\trunk\Apps\Tools

You should install python with default path (ex: C:\Python27), if you did this step before, you skip this step.



Name	Date modified	Type	Size
DLLs	12/5/2014 9:49 AM	File folder	
Doc	12/5/2014 9:50 AM	File folder	
include	12/5/2014 9:49 AM	File folder	
Lib	12/5/2014 9:51 AM	File folder	
libs	12/5/2014 9:49 AM	File folder	
tcl	12/5/2014 9:50 AM	File folder	
Tools	12/5/2014 9:50 AM	File folder	
LICENSE	6/30/2014 4:05 PM	Text Document	38 KB
NEWS	6/30/2014 3:38 PM	Text Document	380 KB
python	6/30/2014 4:03 PM	Application	26 KB
pythonw	6/30/2014 4:04 PM	Application	27 KB
README	6/30/2014 3:37 PM	Text Document	53 KB
w9xpopen	6/30/2014 4:02 PM	Application	49 KB

Figure 3-Python Installed

3.3 Install the Plugin

To install the Plugin, you can find the files (setup.exe, Plugin Installer.msi) in the Installer folder on the SVN at the location:
...\el-external-shc\trunk\Apps\Plugins\ DockablePluginPythonQt_API5\Testing.

a) Click on setup.exe file, the screen will be displayed as below.

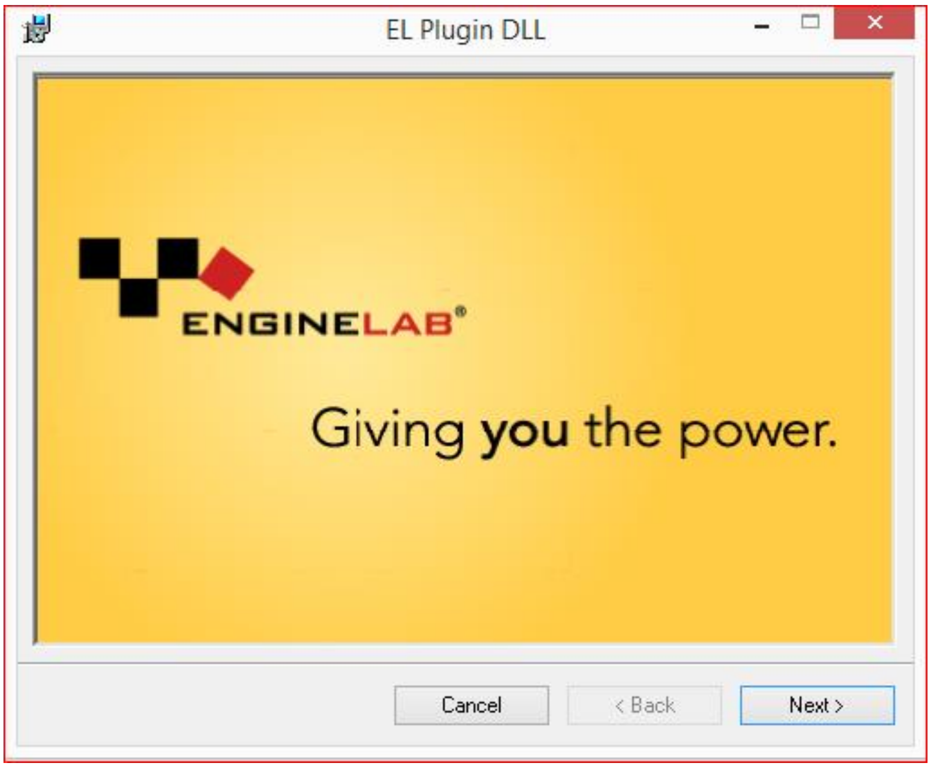
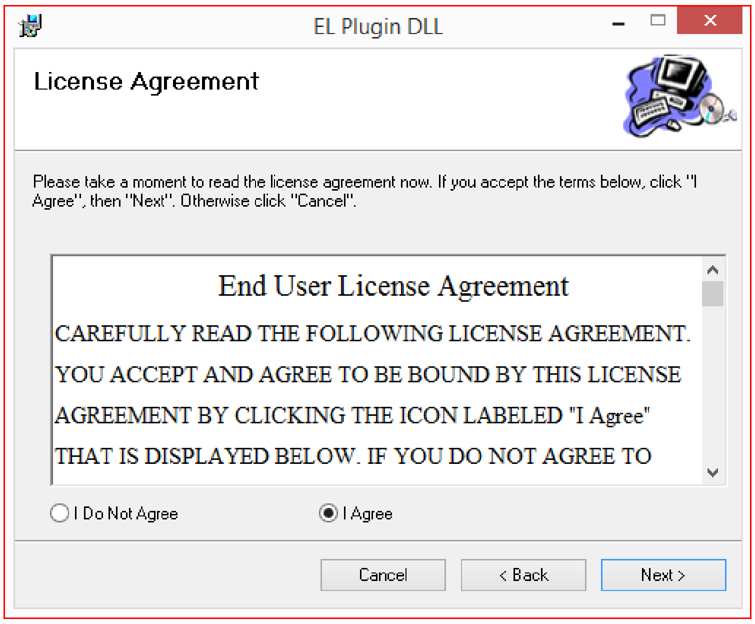


Figure 4-Main Installer

b) Click on Next -> I agree.



License Agreement

Please take a moment to read the license agreement now. If you accept the terms below, click "I Agree", then "Next". Otherwise click "Cancel".

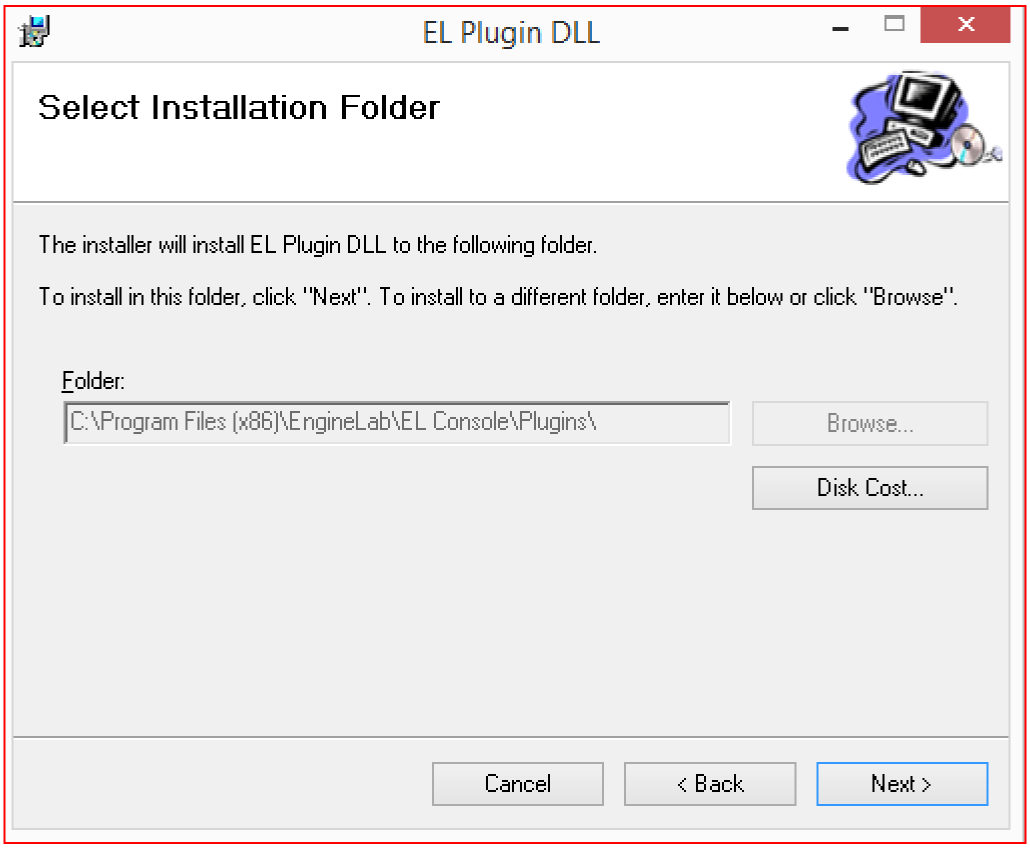
End User License Agreement

CAREFULLY READ THE FOLLOWING LICENSE AGREEMENT. YOU ACCEPT AND AGREE TO BE BOUND BY THIS LICENSE AGREEMENT BY CLICKING THE ICON LABELED "I Agree" THAT IS DISPLAYED BELOW. IF YOU DO NOT AGREE TO

○ I Do Not Agree [✓ ● I Agree]

Figure 5-License Agreement

c) Click on Next, the screen will be displayed as below.



EL Plugin DLL

Select Installation Folder

The installer will install EL Plugin DLL to the following folder.

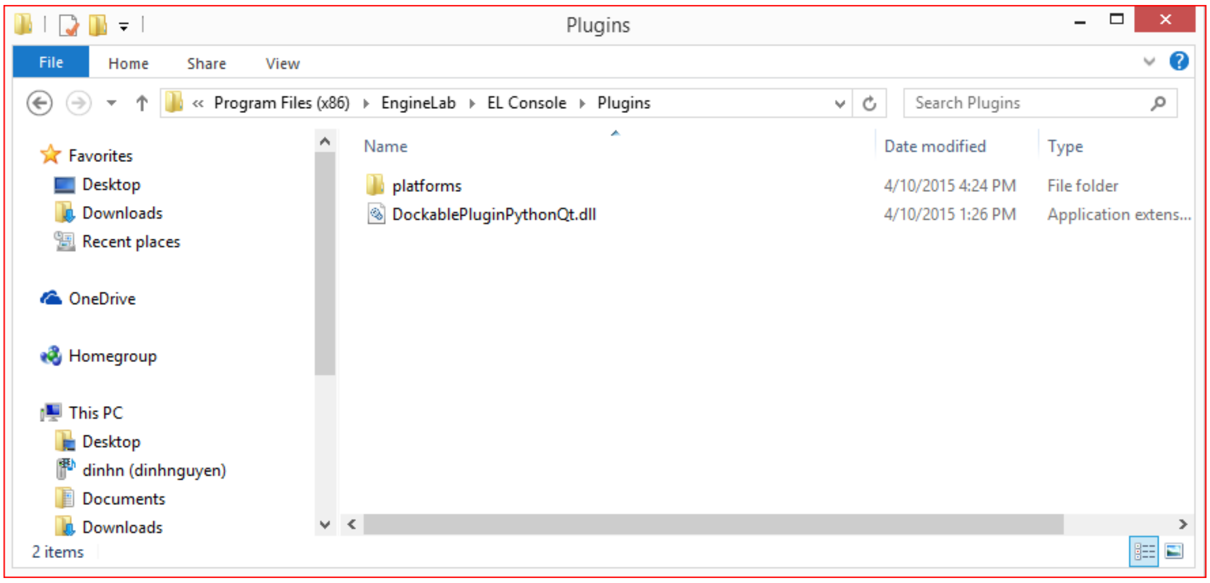
To install in this folder, click "Next". To install to a different folder, enter it below or click "Browse".

Folder: C:\Program Files (x86)\EngineLab\EL Console\Plugins\

Figure 6-Select Installation Folder

d) And Click on Next to complete installing.

e) After install the Plugin, the Plugin dll is placed at the location: C:\Program Files (x86)\EngineLab\EL Console\Plugins



Plugins

« Program Files (x86) ▸ EngineLab ▸ EL Console ▸ Plugins

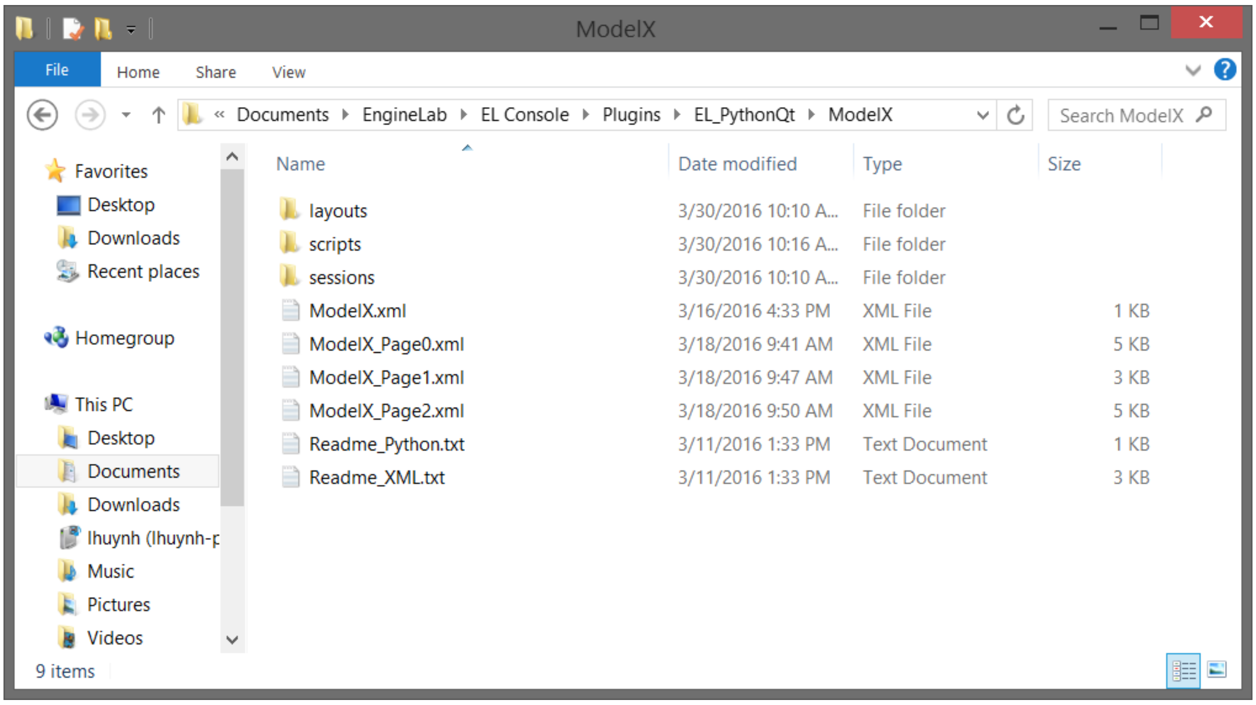
Name	Date modified	Type
platforms	4/10/2015 4:24 PM	File folder
DockablePluginPythonQt.dll	4/10/2015 1:26 PM	Application extens...

2 items

Figure 7-Place of the plugin dll

f) After install the Plugin, the files (.xml, .py) are placed at the location: $(DocumentPath)\EngineLab\EL Console\Plugin\EL_PythonQt

- ➢ The PluginPythonQt.xml file is a XML script. You can modify this file for your testing.
- ➢ The **session** folder contains session file (…"ModelID"\sessions\"ModelID".elcfg) of your model, you should load this file before you start the plugin.
- ➢ The folder Script contains Python script files, each model will have an entry folder which have same name with XML resource file. Also this folder contains a python file "init.py" use to configure Qt control on the first display. For demo purpose, we already created an entry folder named **scripts** and place many sample script file in it.
- ➢ The **layouts** folder contains layout file (…"ModelID"\layouts\"ModelID".ellyt) of your model, you should load this file before you start the plugin.



Name	Date modified	Type	Size
layouts	3/30/2016 10:10 A...	File folder	
scripts	3/30/2016 10:16 A...	File folder	
sessions	3/30/2016 10:10 A...	File folder	
ModelX.xml	3/16/2016 4:33 PM	XML File	1 KB
ModelX_Page0.xml	3/18/2016 9:41 AM	XML File	5 KB
ModelX_Page1.xml	3/18/2016 9:47 AM	XML File	3 KB
ModelX_Page2.xml	3/18/2016 9:50 AM	XML File	5 KB
Readme_Python.txt	3/11/2016 1:33 PM	Text Document	1 KB
Readme_XML.txt	3/11/2016 1:33 PM	Text Document	3 KB

Figure 8-Folder Organization of Plugin

4. Start the demos for testing

4.1 Launch the EL Console aplication

Launch the EL console app, the screen will be displayed as below.

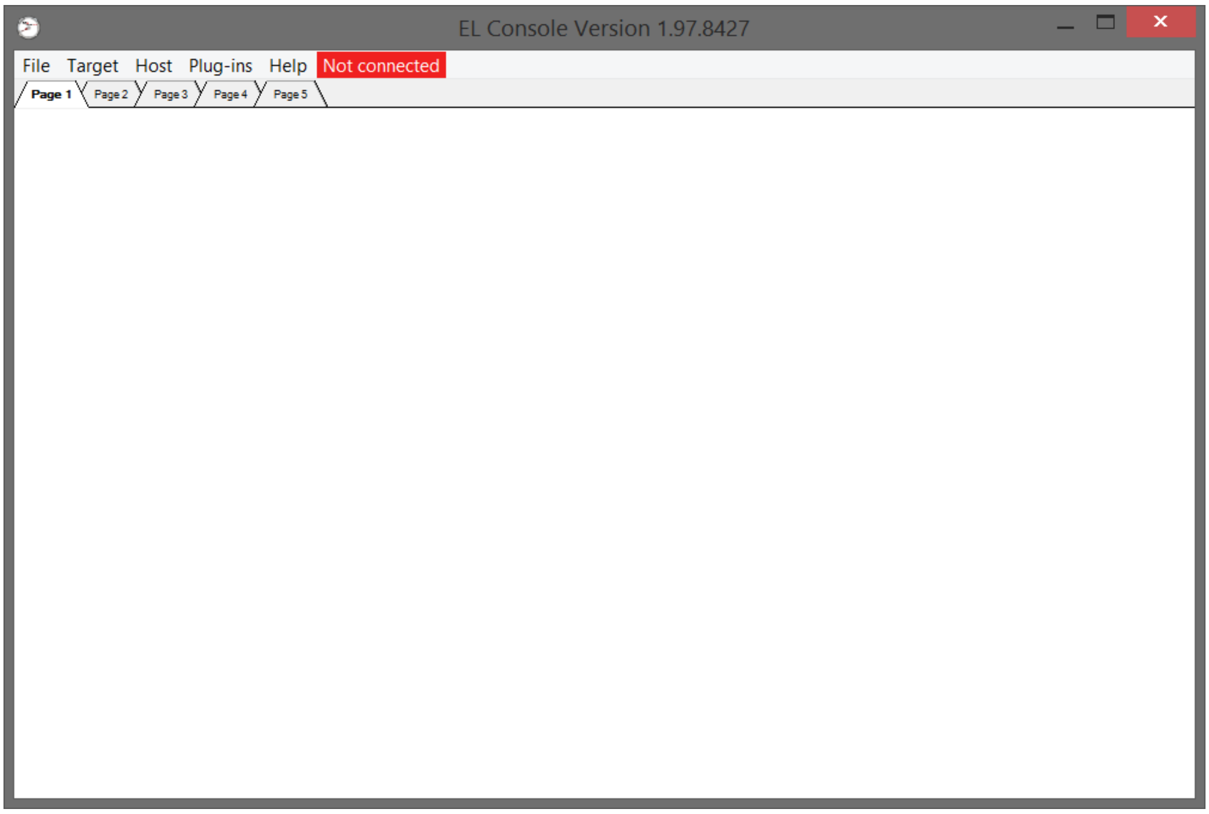
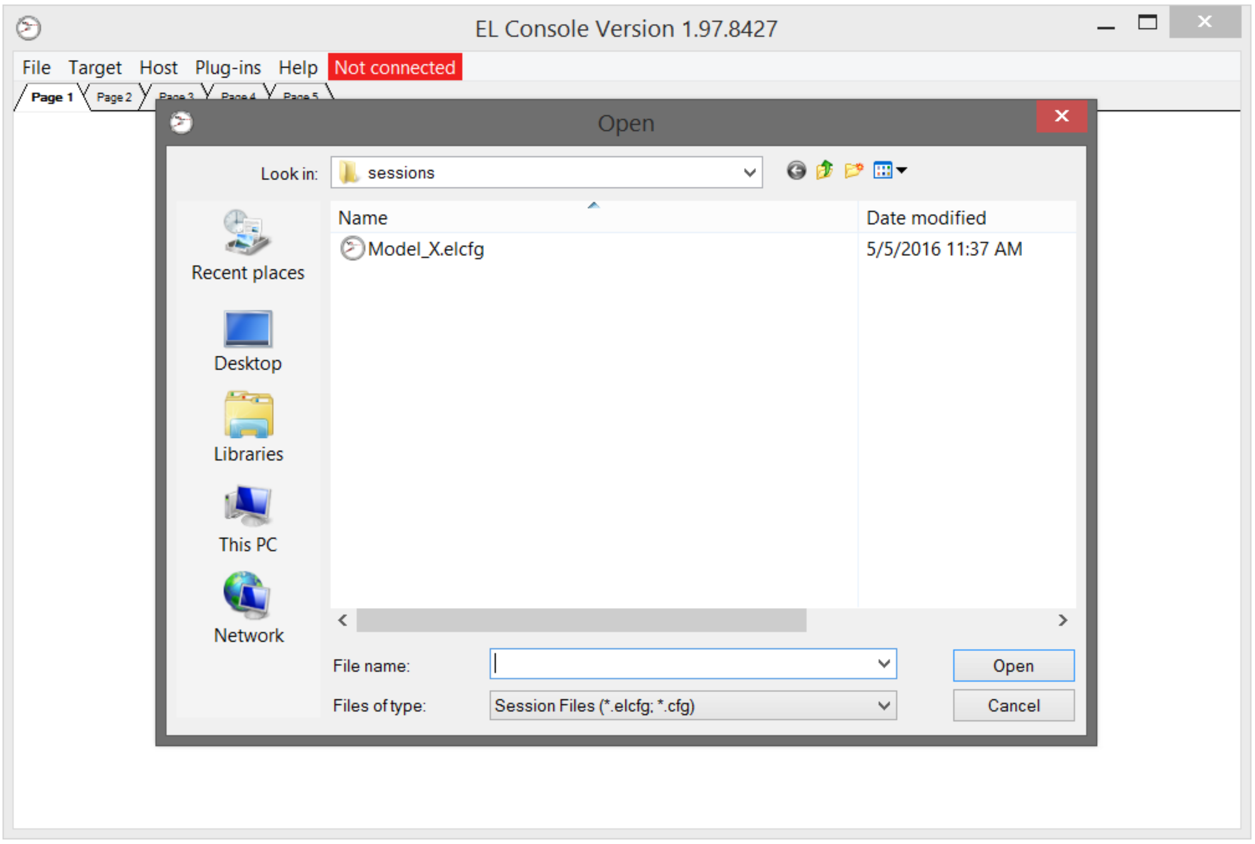


Figure 9-EL Console application

4.2 Testing for ModelX

a) From File menu, select File -> select Open Session -> browse to the Model_X.elcfg file. (Path: $(DocumentPath)\EngineLab\EL Console\Plugins\EL_PythonQt\ModelX\sessions).



File name: Model_X.elcfg — Date modified: 5/5/2016 11:37 AM

Files of type: Session Files (*.elcfg; *.cfg)

Figure 10-Open sample session file

b) From Layout menu, select Host -> Layout -> select Open Layout -> browse to the Layout_ModelX.ellyt file.

(Path: $(DocumentPath)\EngineLab\EL Console\Plugins\EL_PythonQt\ModelX\layouts)

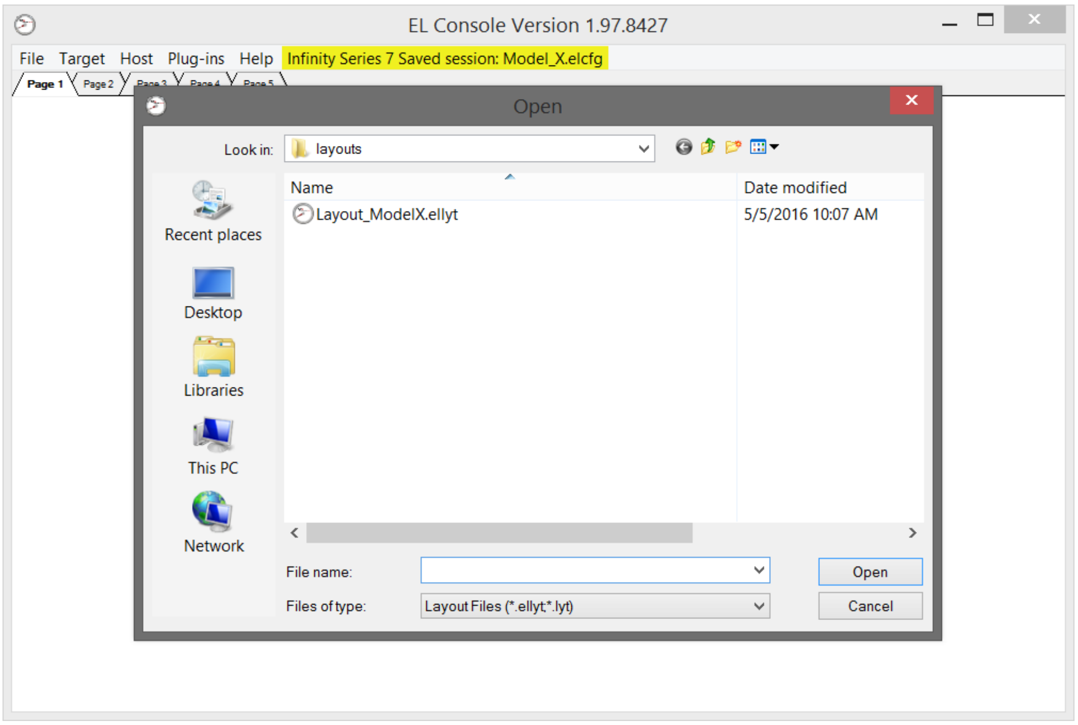
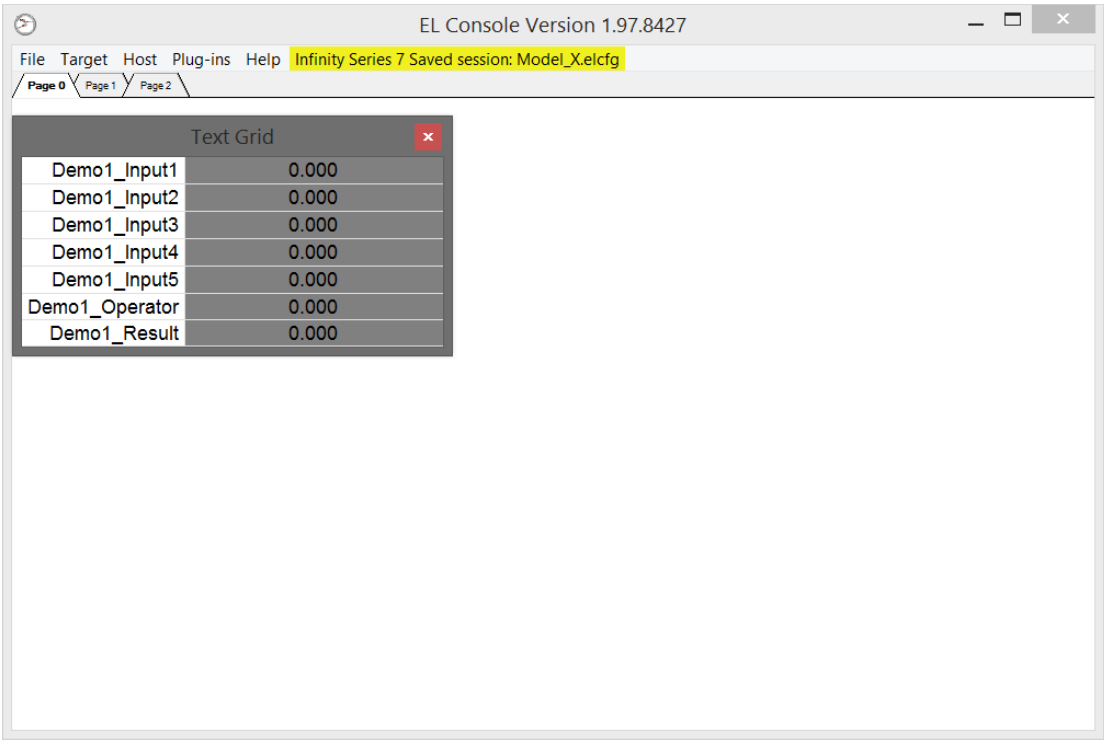


Figure 11-Browse to layout file

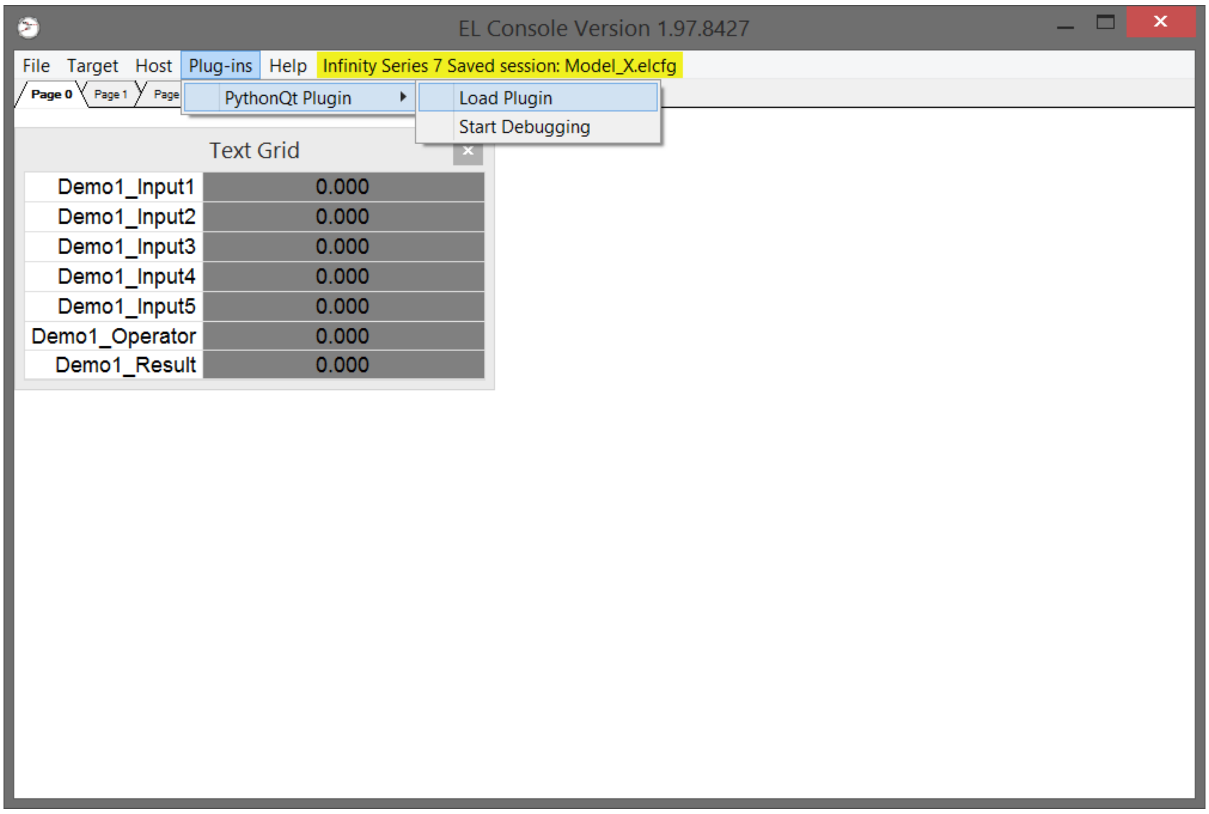
c) After load the layout is successful, the screen is displayed as below.


EL Console Version 1.97.8427
File Target Host Plug-ins Help Infinity Series 7 Saved session: Model_X.elcfg
Page 0 Page 1 Page 2
Text Grid

Demo1_Input1	0.000
Demo1_Input2	0.000
Demo1_Input3	0.000
Demo1_Input4	0.000
Demo1_Input5	0.000
Demo1_Operator	0.000
Demo1_Result	0.000

Figure 12-The screen of Plugin

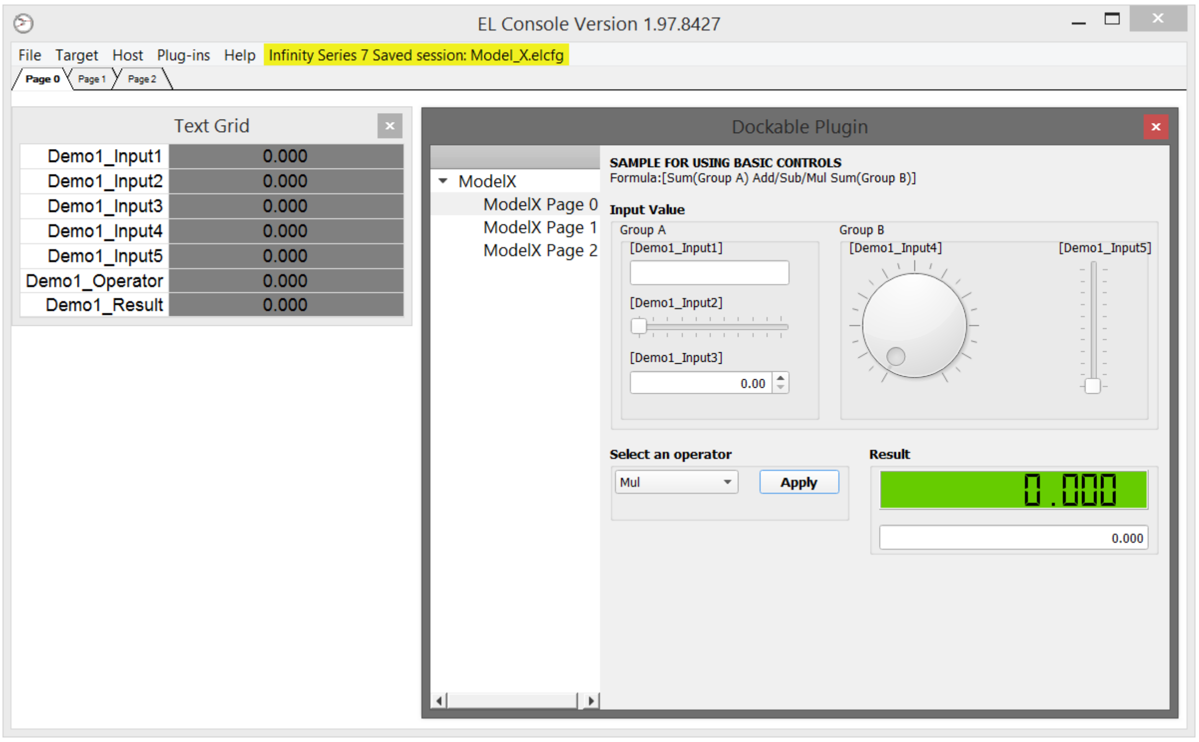
d) Select Plug-ins-> PythonQt Plugin->Load Plugin, the screen will be displayed as below



Demo1_Input1	0.000
Demo1_Input2	0.000
Demo1_Input3	0.000
Demo1_Input4	0.000
Demo1_Input5	0.000
Demo1_Operator	0.000
Demo1_Result	0.000

Figure 13-Load the Plugin

e) After load the Plugin, you can move the main window of the Plugin as below.



Demo1_Input1	0.000
Demo1_Input2	0.000
Demo1_Input3	0.000
Demo1_Input4	0.000
Demo1_Input5	0.000
Demo1_Operator	0.000
Demo1_Result	0.000

SAMPLE FOR USING BASIC CONTROLS
Formula:[Sum(Group A) Add/Sub/Mul Sum(Group B)]

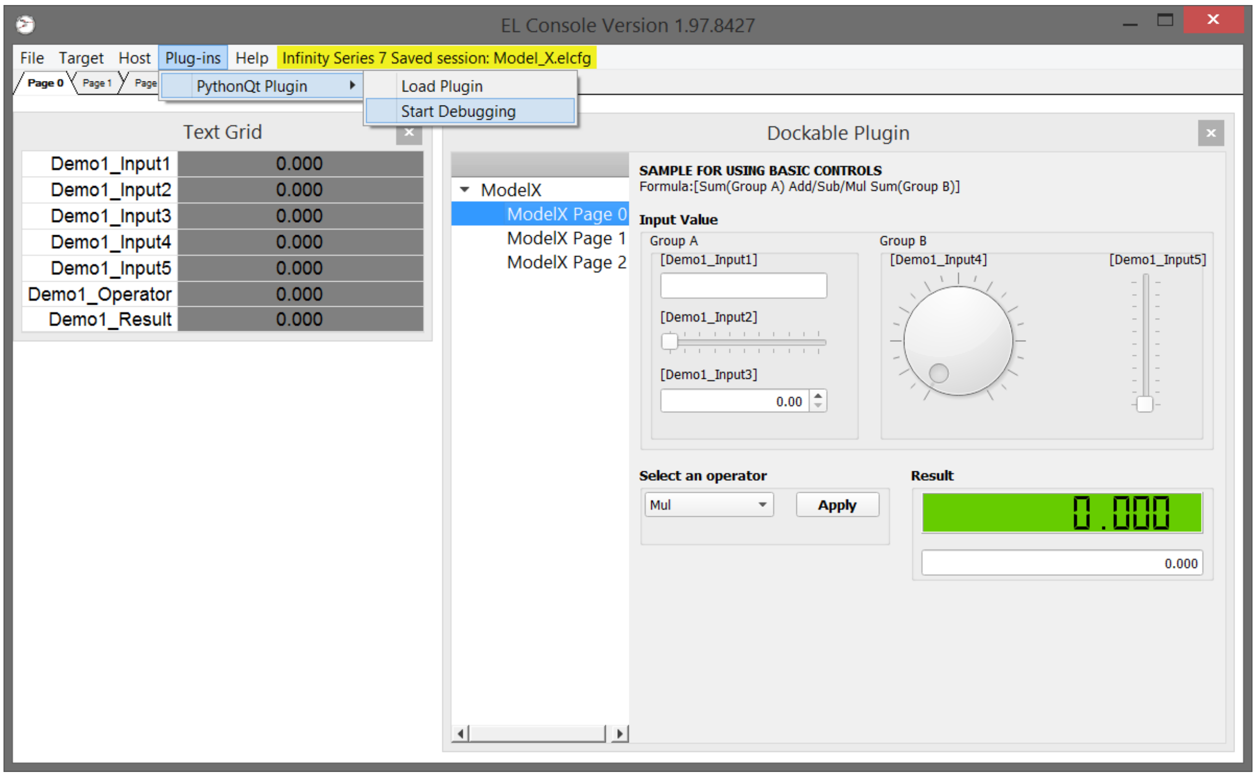
[Demo1_Input3] 0.00

Result: 0.000

0.000

Figure 14- The Plugin

f) Select Plug-ins-> PythonQt Plugin->Start Debugging.

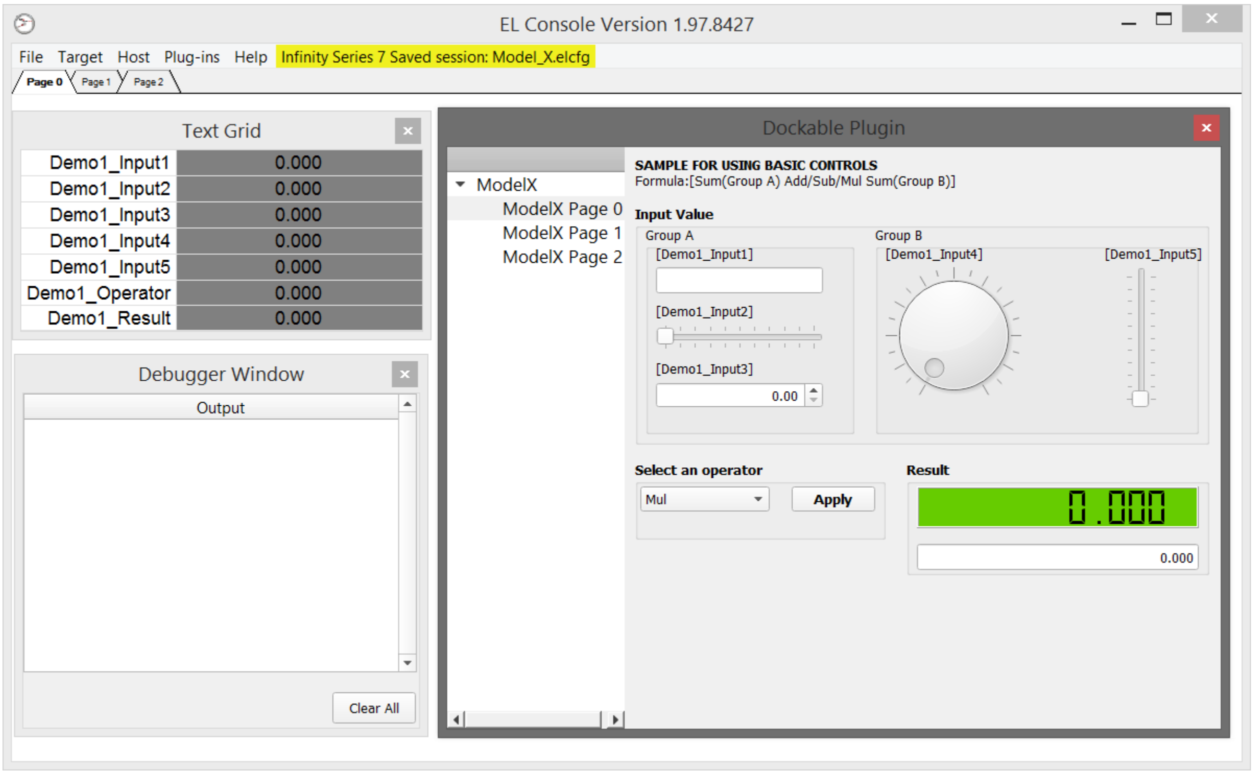


Demo1_Input1	0.000
Demo1_Input2	0.000
Demo1_Input3	0.000
Demo1_Input4	0.000
Demo1_Input5	0.000
Demo1_Operator	0.000
Demo1_Result	0.000

SAMPLE FOR USING BASIC CONTROLS
Formula:[Sum(Group A) Add/Sub/Mul Sum(Group B)]
Input Value
Group A
[Demo1_Input1]
[Demo1_Input2]
[Demo1_Input3] 0.00
Group B
[Demo1_Input4]
[Demo1_Input5]
Select an operator
Result 0.000
0.000

Figure 15- Start the Debugging

g) After load the Debugging, you can move the main window of the Plugin as below.



Demo1_Input1	0.000
Demo1_Input2	0.000
Demo1_Input3	0.000
Demo1_Input4	0.000
Demo1_Input5	0.000
Demo1_Operator	0.000
Demo1_Result	0.000

SAMPLE FOR USING BASIC CONTROLS
Formula:[Sum(Group A) Add/Sub/Mul Sum(Group B)]
Input Value
Group A
[Demo1_Input1]
[Demo1_Input2]
[Demo1_Input3] 0.00
Group B
[Demo1_Input4]
[Demo1_Input5]
Select an operator
Result 0.000
0.000

Figure 16- The Debugging

4.2.1 Testing for ModelX Page 0

For more details, refer to

$(DocumentPath)\ EngineLab\EL Console\Plugins\EL_PythonQt\ModelX\ModelX.xml

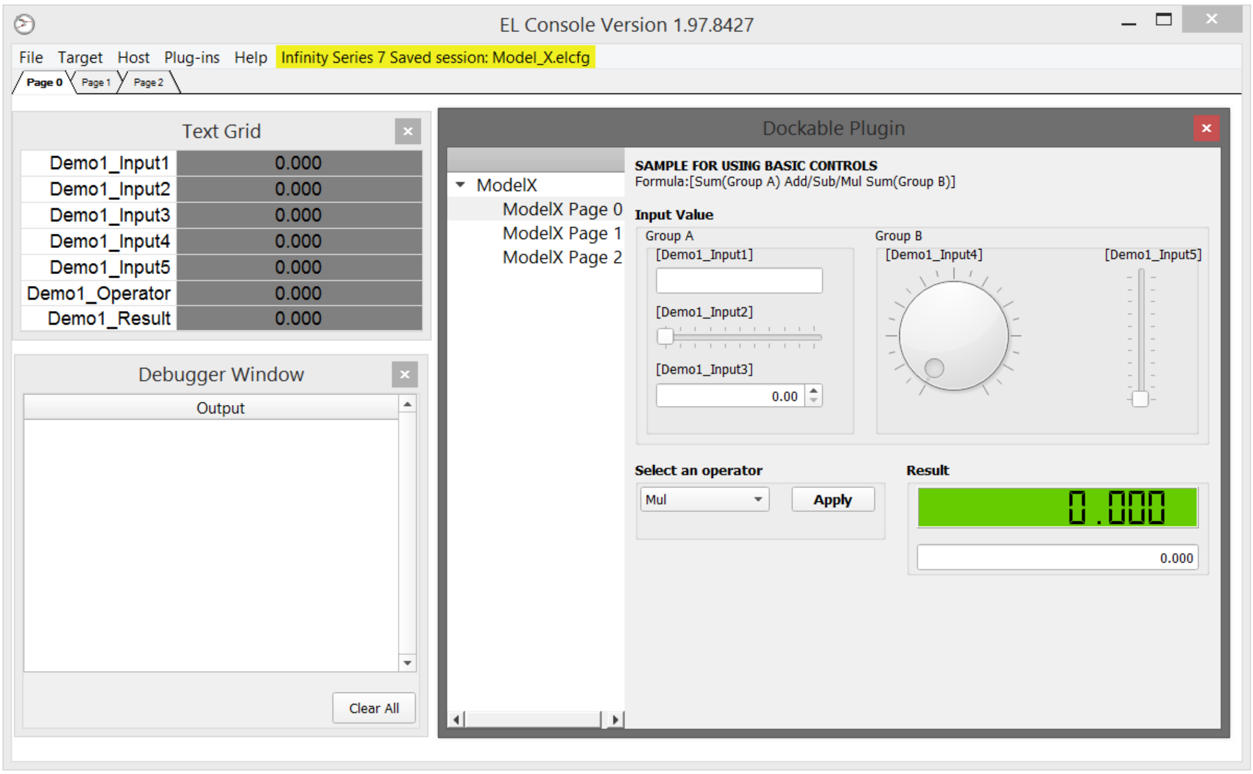
$(DocumentPath)\ EngineLab\EL Console\Plugins\EL_PythonQt\ModelX\ModelX_Page0.xml

$(DocumentPath)\ EngineLab\EL Console\Plugins\EL_PythonQt\ModelX\scripts\Py_ModelX_Page0.py

The demo is example for using the basic controls, so you can change code in XML file or Python code for testing.

a) Set value for the basic controls as picture below.

a) Note: If control is a "read_channel" control, it will render the value of real channel, so if you want to change the value of that control you have to change the value of real channel.



EL Console Version 1.97.8427

File Target Host Plug-ins Help Infinity Series 7 Saved session: Model_X.elcfg

Page 0 Page 1 Page 2

Text Grid

Demo1_Input1	0.000
Demo1_Input2	0.000
Demo1_Input3	0.000
Demo1_Input4	0.000
Demo1_Input5	0.000
Demo1_Operator	0.000
Demo1_Result	0.000

Debugger Window

Output

Clear All

Dockable Plugin

ModelX
- ModelX Page 0
- ModelX Page 1
- ModelX Page 2

SAMPLE FOR USING BASIC CONTROLS

Formula:[Sum(Group A) Add/Sub/Mul Sum(Group B)]

Input Value

Group A

[Demo1_Input1]

[Demo1_Input2]

[Demo1_Input3] 0.00

Group B

[Demo1_Input4]

[Demo1_Input5]

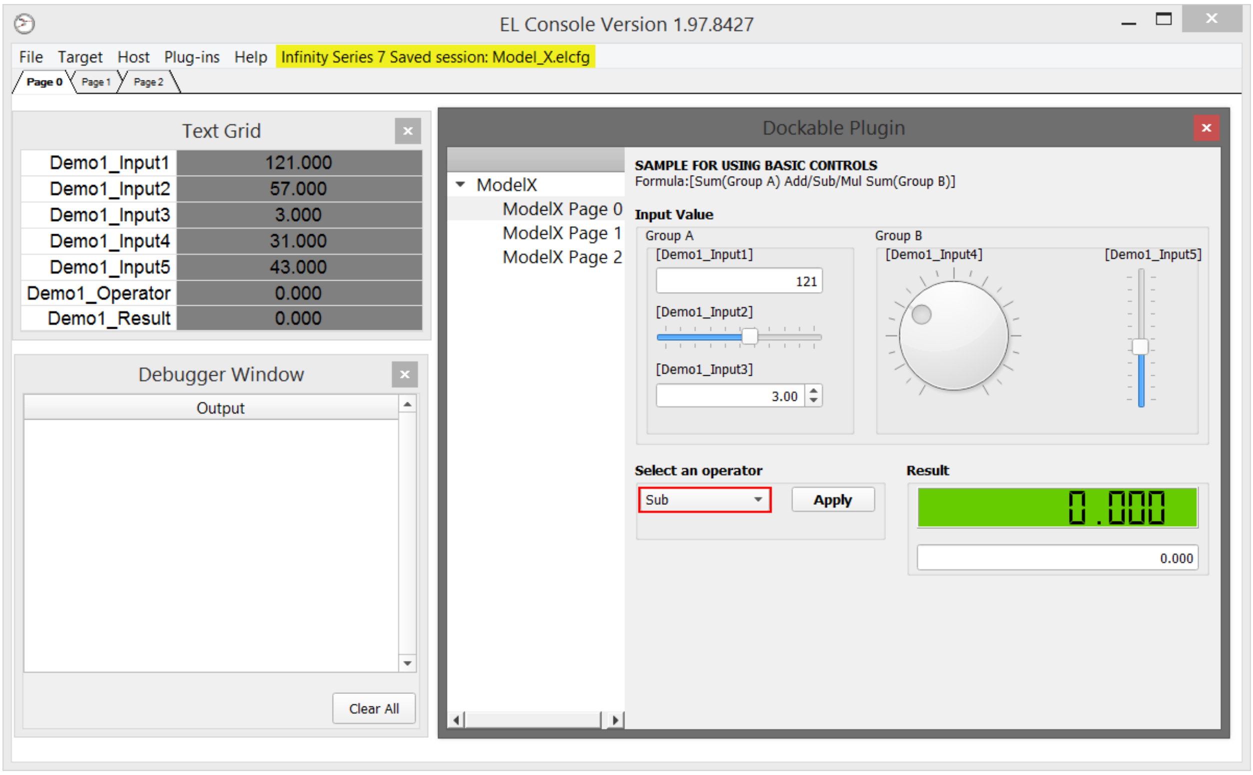
Select an operator: Mul Apply

Result: 0.000

0.000

Figure 17-Setting for the controls

b) Select an operator as picture below.

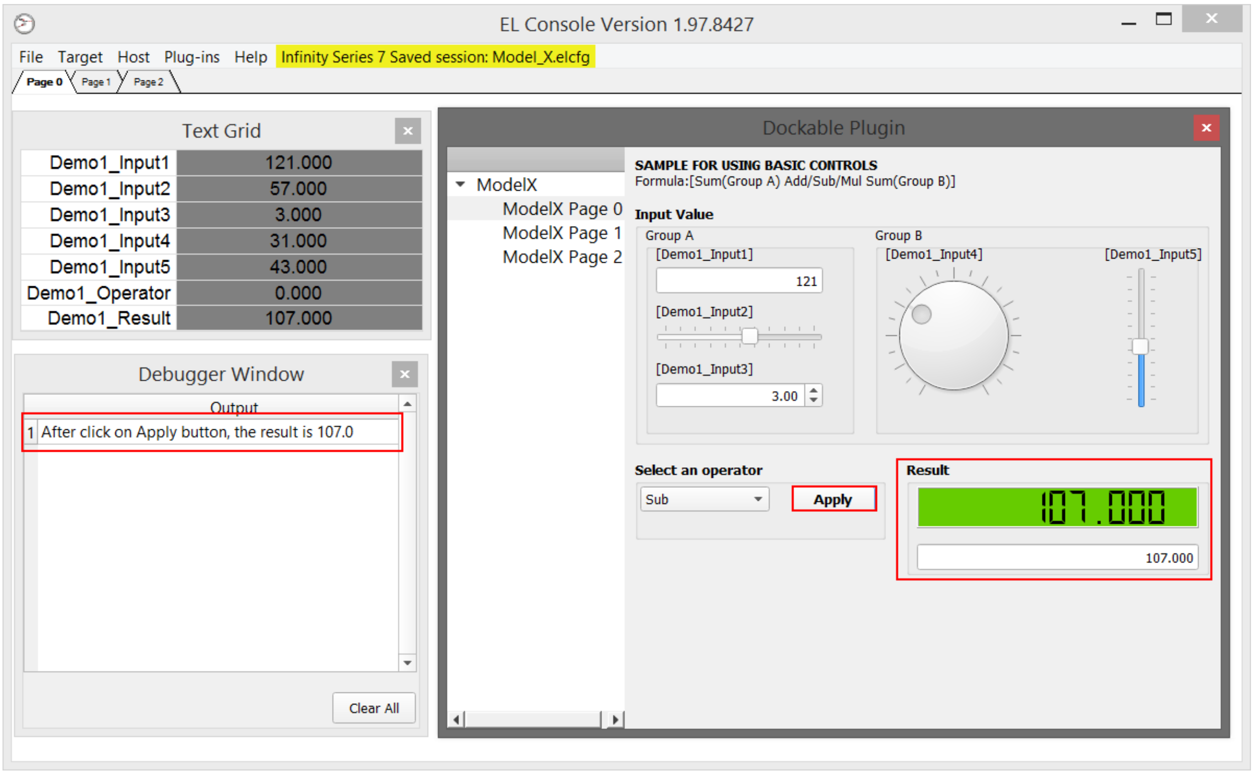


Demo1_Input1	121.000
Demo1_Input2	57.000
Demo1_Input3	3.000
Demo1_Input4	31.000
Demo1_Input5	43.000
Demo1_Operator	0.000
Demo1_Result	0.000

[Demo1_Input1] 121, [Demo1_Input3] 3.00, Result 0.000

Figure 18-Select operator

c) Click on Apply button, the screen is displayed as image below.



Demo1_Input1	121.000
Demo1_Input2	57.000
Demo1_Input3	3.000
Demo1_Input4	31.000
Demo1_Input5	43.000
Demo1_Operator	0.000
Demo1_Result	107.000

```
1 After click on Apply button, the result is 107.0
```

[Demo1_Input1] 121, [Demo1_Input3] 3.00, Result 107.000

Figure 19-The result for testing

4.2.2 Testing for ModelX Page 1

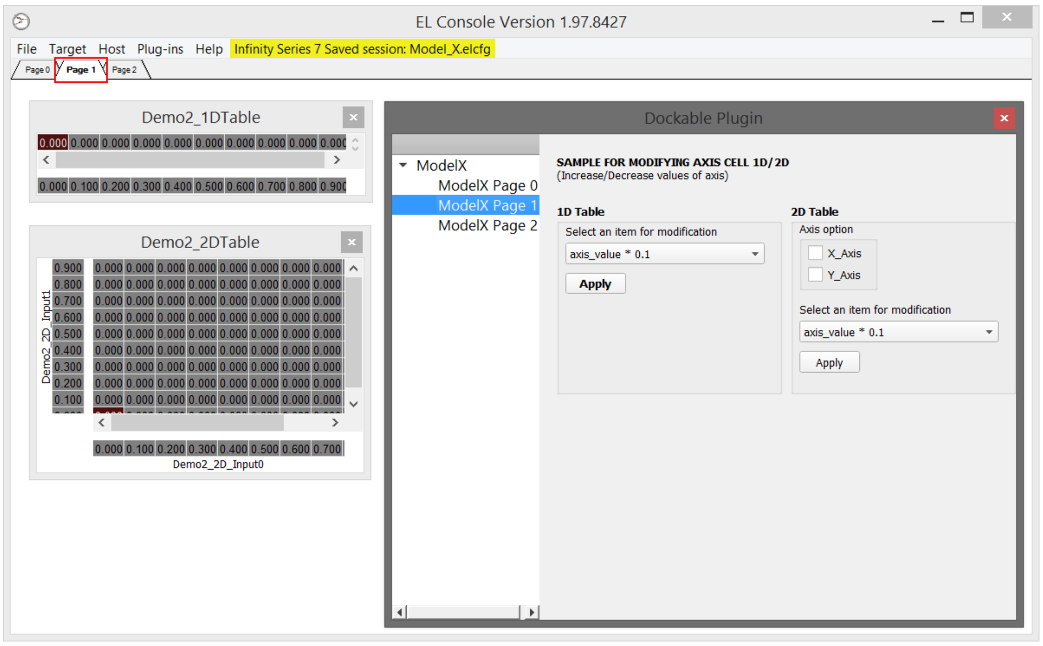
For more details, refer to

$(DocumentPath)\ EngineLab\EL Console\Plugins\EL_PythonQt\ModelX\ModelX.xml

$(DocumentPath)\ EngineLab\EL Console\Plugins\EL_PythonQt\ModelX\ModelX_Page1.xml

$(DocumentPath)\ EngineLab\EL Console\Plugins\EL_PythonQt\ModelX\scripts\Py_ModelX_Page1.py

a) Click on ModelX Page 1 as picture below.



EL Console Version 1.97.8427

File Target Host Plug-ins Help Infinity Series 7 Saved session: Model_X.elcfg

Page 0 Page 1 Page 2

Demo2_1DTable

0.000	0.100	0.200	0.300	0.400	0.500	0.600	0.700	0.800	0.900
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Demo2_2DTable

Demo2_2D_Input1 \ Demo2_2D_Input0	0.000	0.100	0.200	0.300	0.400	0.500	0.600	0.700
0.900	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.800	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.700	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.600	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.500	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.400	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.300	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.200	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.100	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Dockable Plugin

ModelX
- ModelX Page 0
- ModelX Page 1
- ModelX Page 2

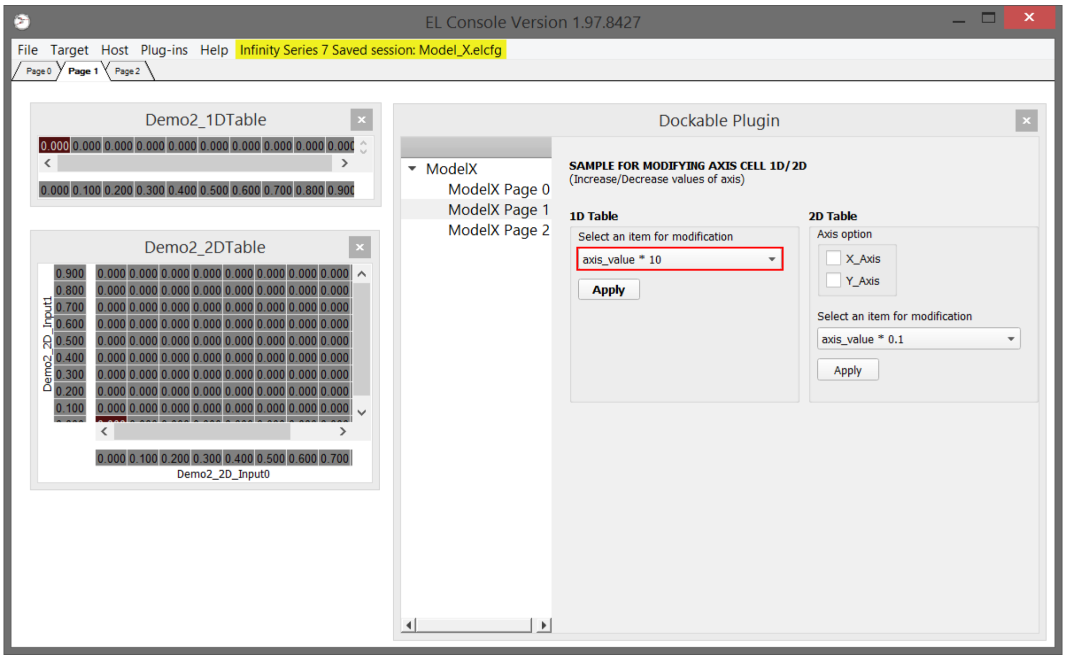
SAMPLE FOR MODIFYING AXIS CELL 1D/2D
(Increase/Decrease values of axis)

1D Table
Select an item for modification: axis_value * 0.1
Apply

2D Table
Axis option: X_Axis, Y_Axis
Select an item for modification: axis_value * 0.1
Apply

Figure 20-Loading ModelX Page 1

b) Increase values of axis cell of 1D Table, select an item as picture below.



EL Console Version 1.97.8427

File Target Host Plug-ins Help Infinity Series 7 Saved session: Model_X.elcfg

Page 0 Page 1 Page 2

Demo2_1DTable

0.000	0.100	0.200	0.300	0.400	0.500	0.600	0.700	0.800	0.900
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Demo2_2DTable

Demo2_2D_Input1 \ Demo2_2D_Input0	0.000	0.100	0.200	0.300	0.400	0.500	0.600	0.700
0.900	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.800	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.700	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.600	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.500	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.400	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.300	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.200	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.100	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Dockable Plugin

ModelX
- ModelX Page 0
- ModelX Page 1
- ModelX Page 2

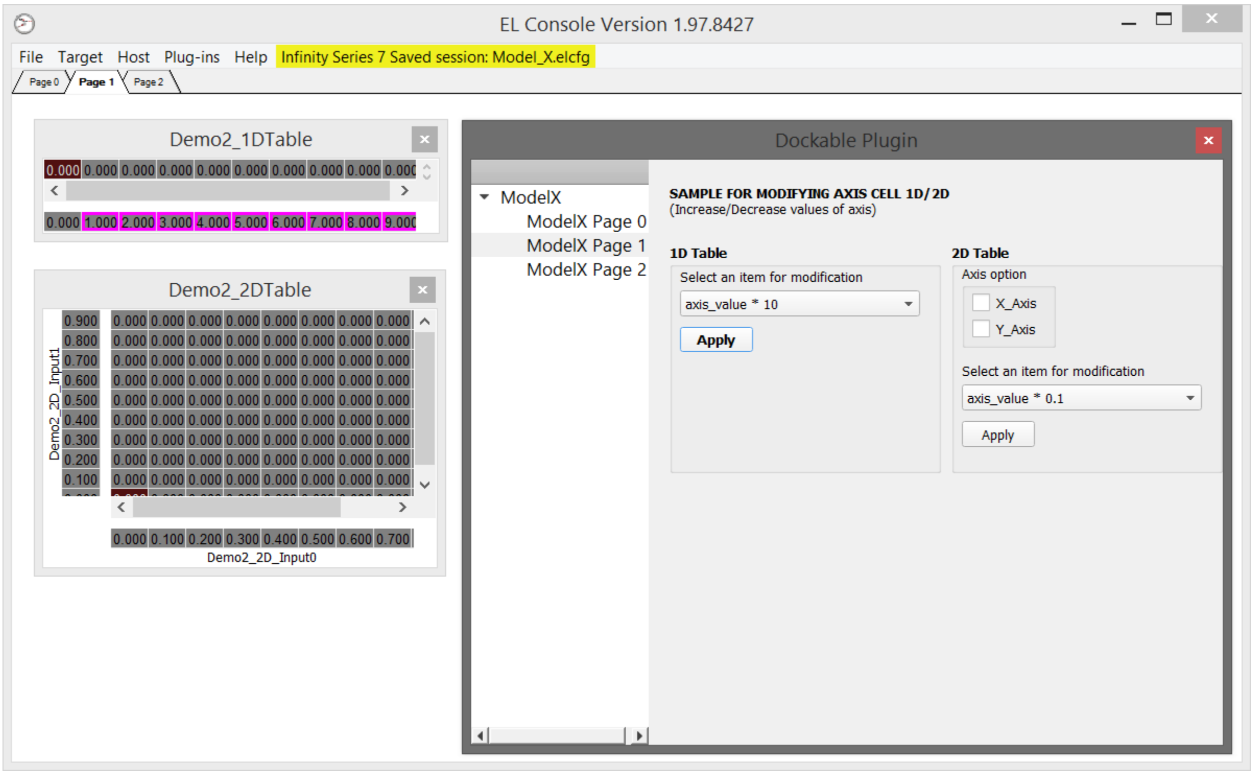
SAMPLE FOR MODIFYING AXIS CELL 1D/2D
(Increase/Decrease values of axis)

1D Table
Select an item for modification: axis_value * 10
Apply

2D Table
Axis option: X_Axis, Y_Axis
Select an item for modification: axis_value * 0.1
Apply

Figure 21-Setting for the controls 1D Table

c) Click on Apply button, check update values as picture below.


EL Console Version 1.97.8427

File Target Host Plug-ins Help Infinity Series 7 Saved session: Model_X.elcfg

Page 0 Page 1 Page 2

Demo2_1DTable

0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	1.000	2.000	3.000	4.000	5.000	6.000	7.000	8.000	9.000

Demo2_2DTable

Demo2_2D_Input1								
0.900	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.800	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.700	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.600	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.500	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.400	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.300	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.200	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.100	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Demo2_2D_Input0	0.000	0.100	0.200	0.300	0.400	0.500	0.600	0.700

Dockable Plugin

ModelX
- ModelX Page 0
- ModelX Page 1
- ModelX Page 2

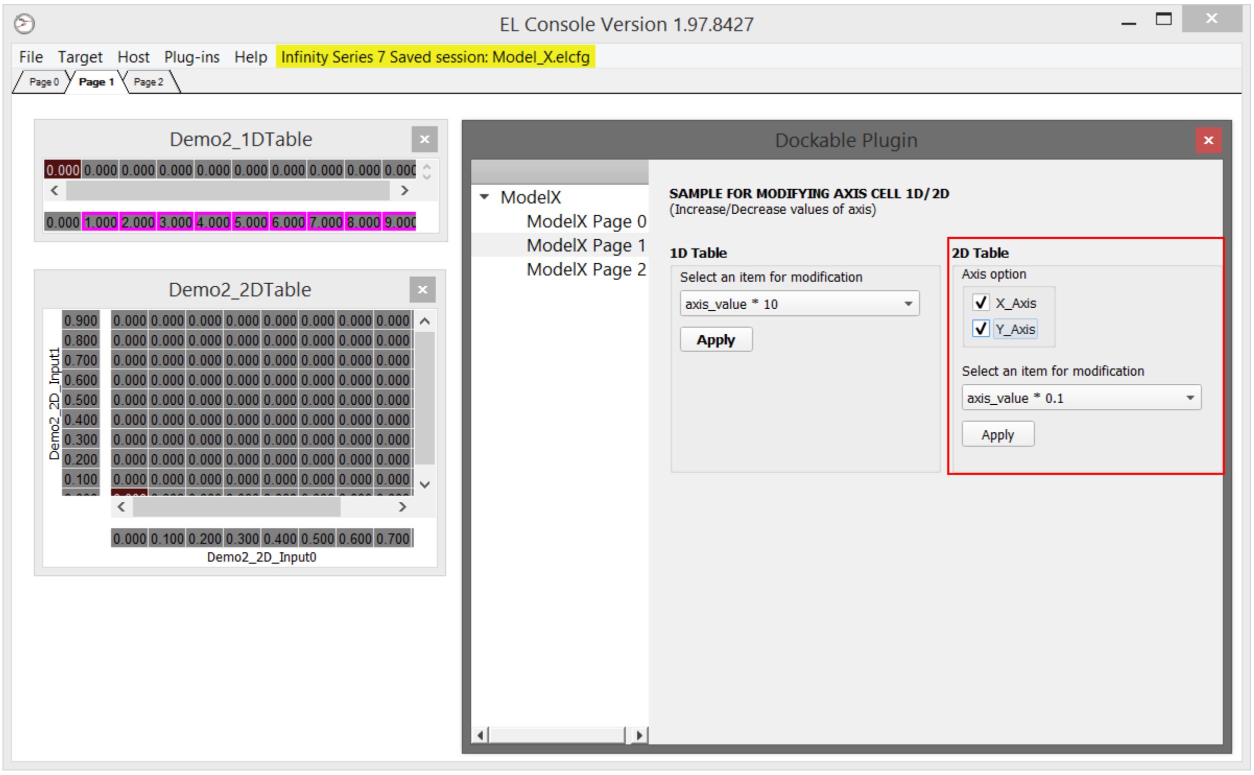
SAMPLE FOR MODIFYING AXIS CELL 1D/2D
(Increase/Decrease values of axis)

1D Table
Select an item for modification: axis_value * 10
Apply

2D Table
Axis option: X_Axis [ ], Y_Axis [ ]
Select an item for modification: axis_value * 0.1
Apply

Figure 22-The result for testing 1D Table

d) Increase values of axis cell of 2D Table, setting for the controls as picture below.


EL Console Version 1.97.8427

File Target Host Plug-ins Help Infinity Series 7 Saved session: Model_X.elcfg

Page 0 Page 1 Page 2

Demo2_1DTable

0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	1.000	2.000	3.000	4.000	5.000	6.000	7.000	8.000	9.000

Demo2_2DTable

Demo2_2D_Input1								
0.900	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.800	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.700	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.600	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.500	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.400	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.300	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.200	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.100	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Demo2_2D_Input0	0.000	0.100	0.200	0.300	0.400	0.500	0.600	0.700

Dockable Plugin

ModelX
- ModelX Page 0
- ModelX Page 1
- ModelX Page 2

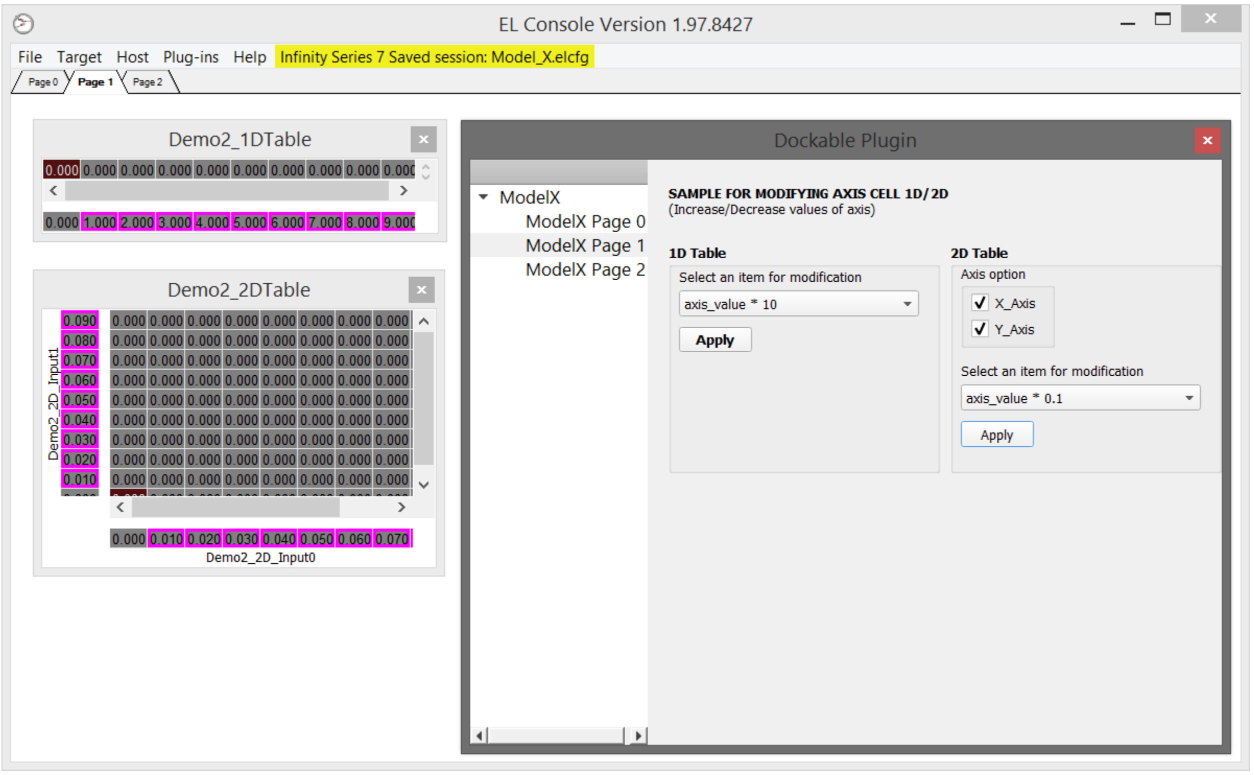
SAMPLE FOR MODIFYING AXIS CELL 1D/2D
(Increase/Decrease values of axis)

1D Table
Select an item for modification: axis_value * 10
Apply

2D Table
Axis option: [✓ X_Axis], [✓ Y_Axis]
Select an item for modification: axis_value * 0.1
Apply

Figure 23-Setting for the controls 2D Table

e) Click on Apply button, check update values as picture below.



EL Console Version 1.97.8427

File Target Host Plug-ins Help Infinity Series 7 Saved session: Model_X.elcfg

Page 0 Page 1 Page 2

Demo2_1DTable

0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.00
0.000	1.000	2.000	3.000	4.000	5.000	6.000	7.000	8.000	9.00

Demo2_2DTable

Demo2_2D_Input1								
0.090	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.080	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.070	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.060	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.050	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.040	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.030	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.020	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.010	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	0.000	0.010	0.020	0.030	0.040	0.050	0.060	0.070

Demo2_2D_Input0

Dockable Plugin

- ModelX
  - ModelX Page 0
  - ModelX Page 1
  - ModelX Page 2

SAMPLE FOR MODIFYING AXIS CELL 1D/2D
(Increase/Decrease values of axis)

1D Table
Select an item for modification: axis_value * 10
Apply

2D Table
Axis option
✓ X_Axis
✓ Y_Axis
Select an item for modification: axis_value * 0.1
Apply

Figure 24-The result for testing 2D Table

4.2.3 Testing for ModelX Page 2

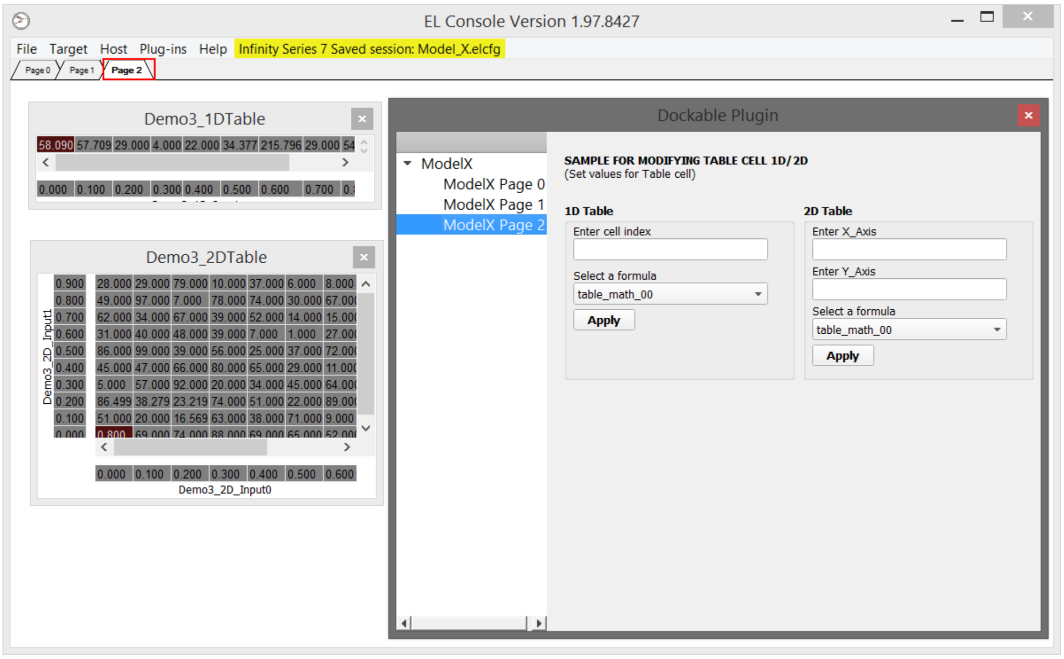
For more details, refer to

$(DocumentPath)\ EngineLab\EL Console\Plugins\EL_PythonQt\ModelX\ModelX.xml

$(DocumentPath)\ EngineLab\EL Console\Plugins\EL_PythonQt\ModelX\ModelX_Page2.xml

$(DocumentPath)\ EngineLab\EL Console\Plugins\EL_PythonQt\ModelX\scripts\Py_ModelX_Page2.py

a) Click on ModelX Page 2 as picture below.



Demo3_1DTable

58.090	57.709	29.000	4.000	22.000	34.377	215.796	29.000	54
0.000	0.100	0.200	0.300	0.400	0.500	0.600	0.700	0.

Demo3_2DTable

Demo3_2D_Input1							
0.900	28.000	29.000	79.000	10.000	37.000	6.000	8.000
0.800	49.000	97.000	7.000	78.000	74.000	30.000	67.00
0.700	62.000	34.000	67.000	39.000	52.000	14.000	15.00
0.600	31.000	40.000	48.000	39.000	7.000	1.000	27.00
0.500	86.000	99.000	39.000	56.000	25.000	37.000	72.00
0.400	45.000	47.000	66.000	80.000	65.000	29.000	11.00
0.300	5.000	57.000	92.000	20.000	34.000	45.000	64.00
0.200	86.499	38.279	23.219	74.000	51.000	22.000	89.00
0.100	51.000	20.000	16.569	63.000	38.000	71.000	9.000
0.000	0.800	69.000	74.000	88.000	69.000	65.000	52.00
Demo3_2D_Input0	0.000	0.100	0.200	0.300	0.400	0.500	0.600

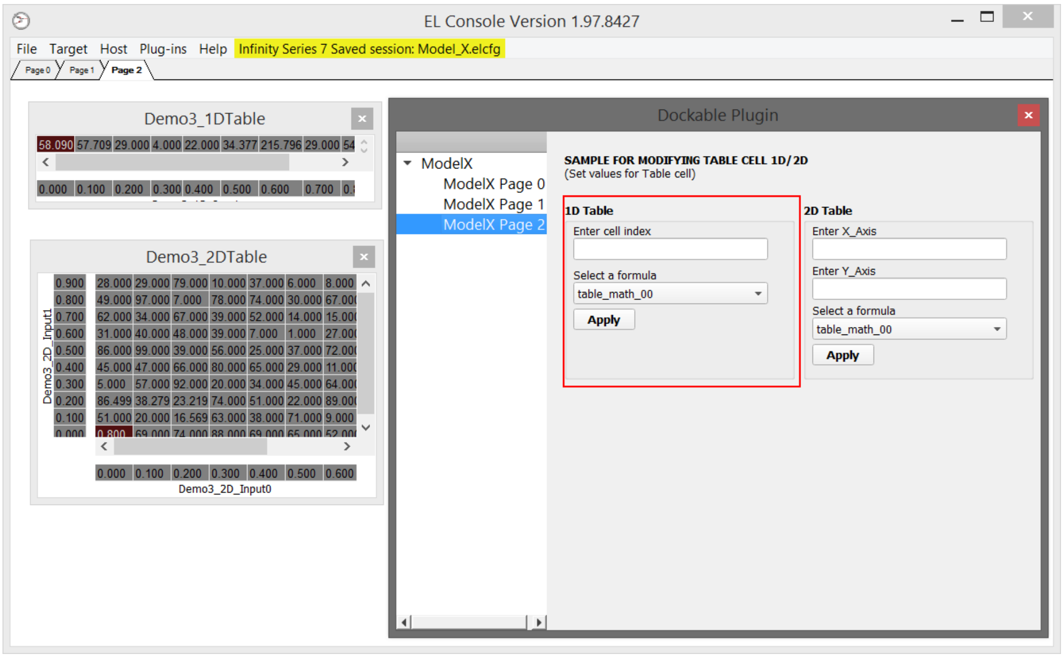
SAMPLE FOR MODIFYING TABLE CELL 1D/2D
(Set values for Table cell)

1D Table
Enter cell index:
Select a formula: table_math_00
Apply

2D Table
Enter X_Axis:
Enter Y_Axis:
Select a formula: table_math_00
Apply

Figure 25-Loading ModelX Page 2

b) Set values for Table cell 1D, setting for the controls as picture below.



Demo3_1DTable

58.090	57.709	29.000	4.000	22.000	34.377	215.796	29.000	54
0.000	0.100	0.200	0.300	0.400	0.500	0.600	0.700	0.

Demo3_2DTable

Demo3_2D_Input1							
0.900	28.000	29.000	79.000	10.000	37.000	6.000	8.000
0.800	49.000	97.000	7.000	78.000	74.000	30.000	67.00
0.700	62.000	34.000	67.000	39.000	52.000	14.000	15.00
0.600	31.000	40.000	48.000	39.000	7.000	1.000	27.00
0.500	86.000	99.000	39.000	56.000	25.000	37.000	72.00
0.400	45.000	47.000	66.000	80.000	65.000	29.000	11.00
0.300	5.000	57.000	92.000	20.000	34.000	45.000	64.00
0.200	86.499	38.279	23.219	74.000	51.000	22.000	89.00
0.100	51.000	20.000	16.569	63.000	38.000	71.000	9.000
0.000	0.800	69.000	74.000	88.000	69.000	65.000	52.00
Demo3_2D_Input0	0.000	0.100	0.200	0.300	0.400	0.500	0.600

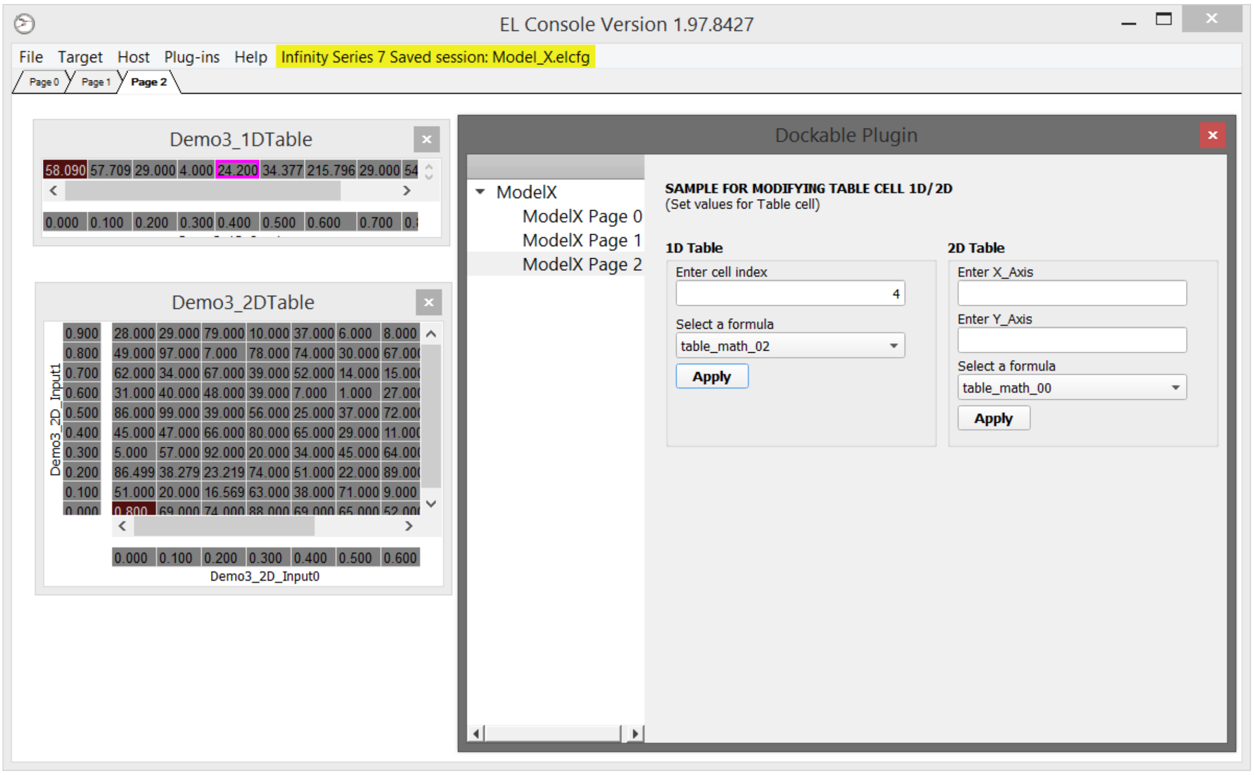
SAMPLE FOR MODIFYING TABLE CELL 1D/2D
(Set values for Table cell)

1D Table
Enter cell index:
Select a formula: table_math_00
Apply

2D Table
Enter X_Axis:
Enter Y_Axis:
Select a formula: table_math_00
Apply

Figure 26-Setting for the controls 1D Table

c) Click on Apply button, check update value as picture below.



Demo3_1DTable

58.090	57.709	29.000	4.000	24.200	34.377	215.796	29.000	54
0.000	0.100	0.200	0.300	0.400	0.500	0.600	0.700	0.

Demo3_2DTable

Demo3_2D_Input1							
0.900	28.000	29.000	79.000	10.000	37.000	6.000	8.000
0.800	49.000	97.000	7.000	78.000	74.000	30.000	67.00
0.700	62.000	34.000	67.000	39.000	52.000	14.000	15.00
0.600	31.000	40.000	48.000	39.000	7.000	1.000	27.00
0.500	86.000	99.000	39.000	56.000	25.000	37.000	72.00
0.400	45.000	47.000	66.000	80.000	65.000	29.000	11.00
0.300	5.000	57.000	92.000	20.000	34.000	45.000	64.00
0.200	86.499	38.279	23.219	74.000	51.000	22.000	89.00
0.100	51.000	20.000	16.569	63.000	38.000	71.000	9.000
0.000	0.800	69.000	74.000	88.000	69.000	65.000	52.00
Demo3_2D_Input0	0.000	0.100	0.200	0.300	0.400	0.500	0.600

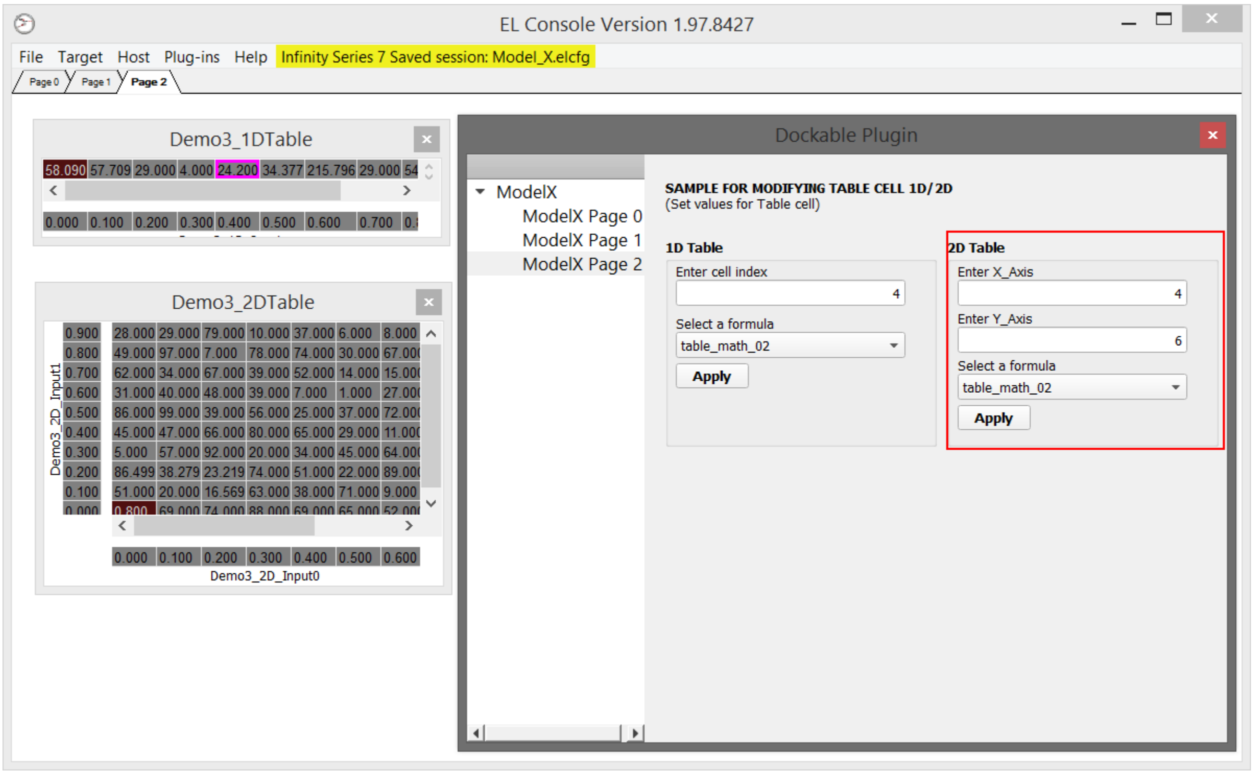
SAMPLE FOR MODIFYING TABLE CELL 1D/2D
(Set values for Table cell)

1D Table
Enter cell index: 4
Select a formula: table_math_02
Apply

2D Table
Enter X_Axis:
Enter Y_Axis:
Select a formula: table_math_00
Apply

Figure 27-The result for testing 1D Table

d) Set values for Table cell 2D, setting for the controls as picture below.



Demo3_1DTable

58.090	57.709	29.000	4.000	24.200	34.377	215.796	29.000	54
0.000	0.100	0.200	0.300	0.400	0.500	0.600	0.700	0.

Demo3_2DTable

Demo3_2D_Input1							
0.900	28.000	29.000	79.000	10.000	37.000	6.000	8.000
0.800	49.000	97.000	7.000	78.000	74.000	30.000	67.00
0.700	62.000	34.000	67.000	39.000	52.000	14.000	15.00
0.600	31.000	40.000	48.000	39.000	7.000	1.000	27.00
0.500	86.000	99.000	39.000	56.000	25.000	37.000	72.00
0.400	45.000	47.000	66.000	80.000	65.000	29.000	11.00
0.300	5.000	57.000	92.000	20.000	34.000	45.000	64.00
0.200	86.499	38.279	23.219	74.000	51.000	22.000	89.00
0.100	51.000	20.000	16.569	63.000	38.000	71.000	9.000
0.000	0.800	69.000	74.000	88.000	69.000	65.000	52.00
Demo3_2D_Input0	0.000	0.100	0.200	0.300	0.400	0.500	0.600

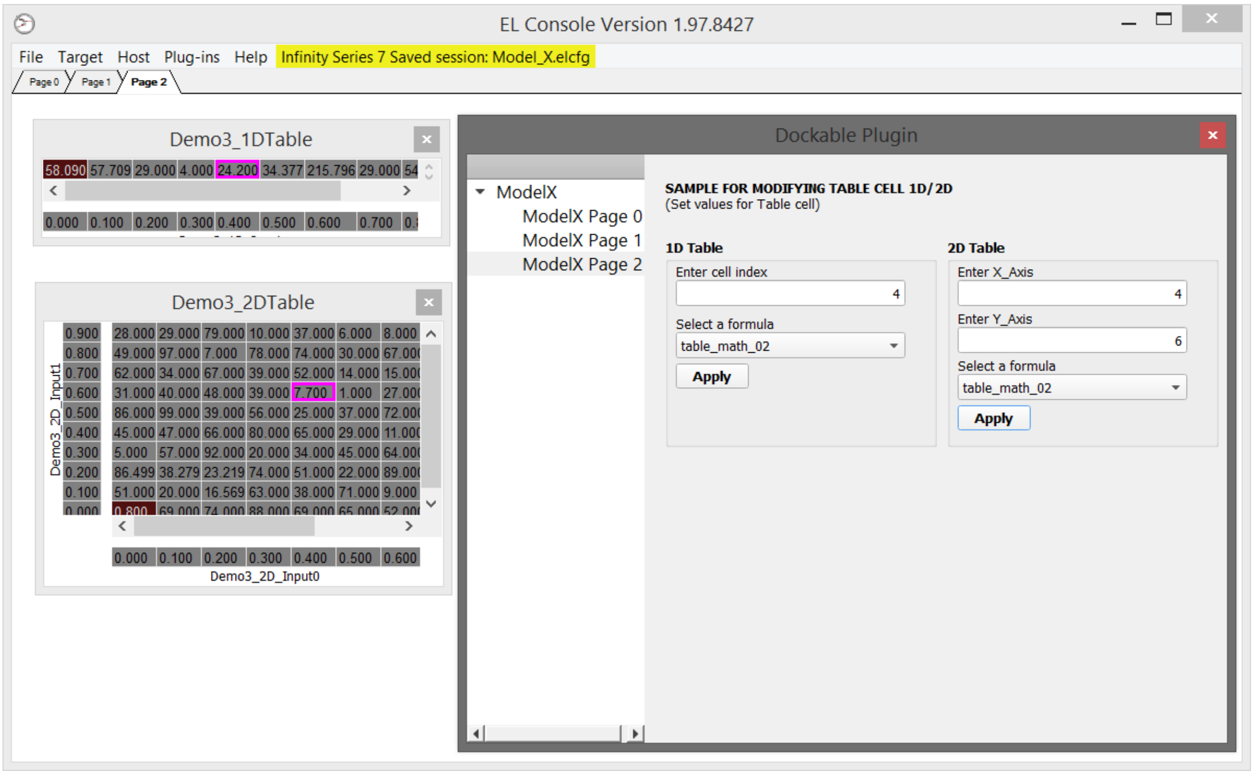
SAMPLE FOR MODIFYING TABLE CELL 1D/2D
(Set values for Table cell)

1D Table
Enter cell index: 4
Select a formula: table_math_02
Apply

2D Table
Enter X_Axis: 4
Enter Y_Axis: 6
Select a formula: table_math_02
Apply

Figure 28-Setting for the controls 2D Table

e) Click on Apply button, check update value as picture below.


EL Console Version 1.97.8427
File Target Host Plug-ins Help
Infinity Series 7 Saved session: Model_X.elcfg
Page 0
Page 1
Page 2
Dockable Plugin
ModelX
ModelX Page 0
ModelX Page 1
ModelX Page 2
Apply
Apply

Demo3_1DTable

58.090	57.709	29.000	4.000	24.200	34.377	215.796	29.000	54
0.000	0.100	0.200	0.300	0.400	0.500	0.600	0.700	0.

Demo3_2DTable

Demo3_2D_Input1							
0.900	28.000	29.000	79.000	10.000	37.000	6.000	8.000
0.800	49.000	97.000	7.000	78.000	74.000	30.000	67.00
0.700	62.000	34.000	67.000	39.000	52.000	14.000	15.00
0.600	31.000	40.000	48.000	39.000	7.700	1.000	27.00
0.500	86.000	99.000	39.000	56.000	25.000	37.000	72.00
0.400	45.000	47.000	66.000	80.000	65.000	29.000	11.00
0.300	5.000	57.000	92.000	20.000	34.000	45.000	64.00
0.200	86.499	38.279	23.219	74.000	51.000	22.000	89.00
0.100	51.000	20.000	16.569	63.000	38.000	71.000	9.000
0.000	0.800	69.000	74.000	88.000	69.000	65.000	52.00
	0.000	0.100	0.200	0.300	0.400	0.500	0.600

Demo3_2D_Input0

SAMPLE FOR MODIFYING TABLE CELL 1D/2D
(Set values for Table cell)

1D Table

Enter cell index: 4

Select a formula: table_math_02

2D Table

Enter X_Axis: 4

Enter Y_Axis: 6

Select a formula: table_math_02

Figure 29-The result for testing 2D Table

4.3 Testing for ModelY

a) From File menu, select File -> select Open Session -> browse to the Model_Y.elcfg file. (Path: $(DocumentPath)\EngineLab\EL Console\Plugins\EL_PythonQt\ModelY\sessions).

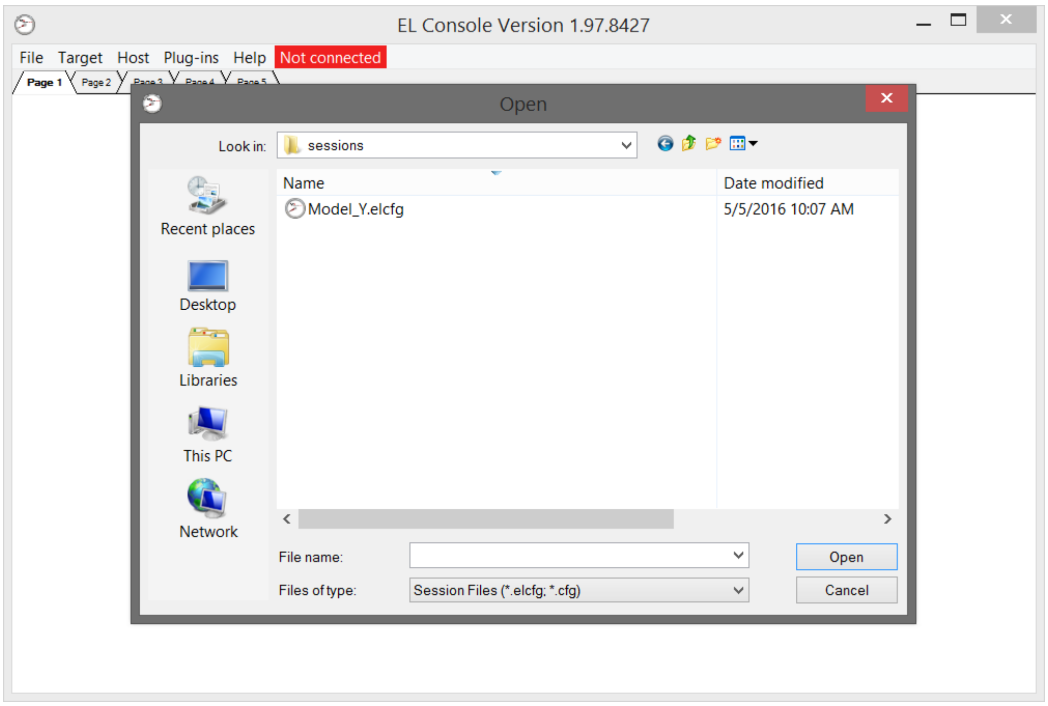


Figure 30-Open sample session file

b) Select Plug-ins-> PythonQt Plugin->Load Plugin, the screen will be displayed as below

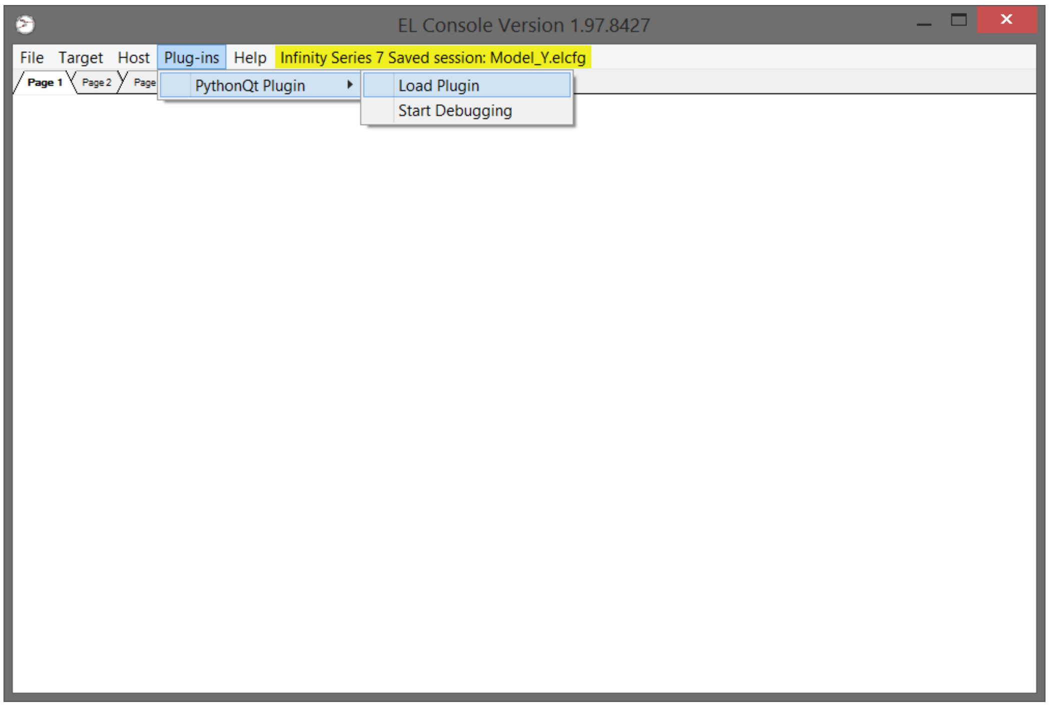

EL Console Version 1.97.8427
File
Target
Host
Plug-ins
Help
Infinity Series 7 Saved session: Model_Y.elcfg
PythonQt Plugin
Load Plugin
Start Debugging

Figure 31-Load the Plugin

c) After load the Plugin, you can move the main window of the Plugin as below.

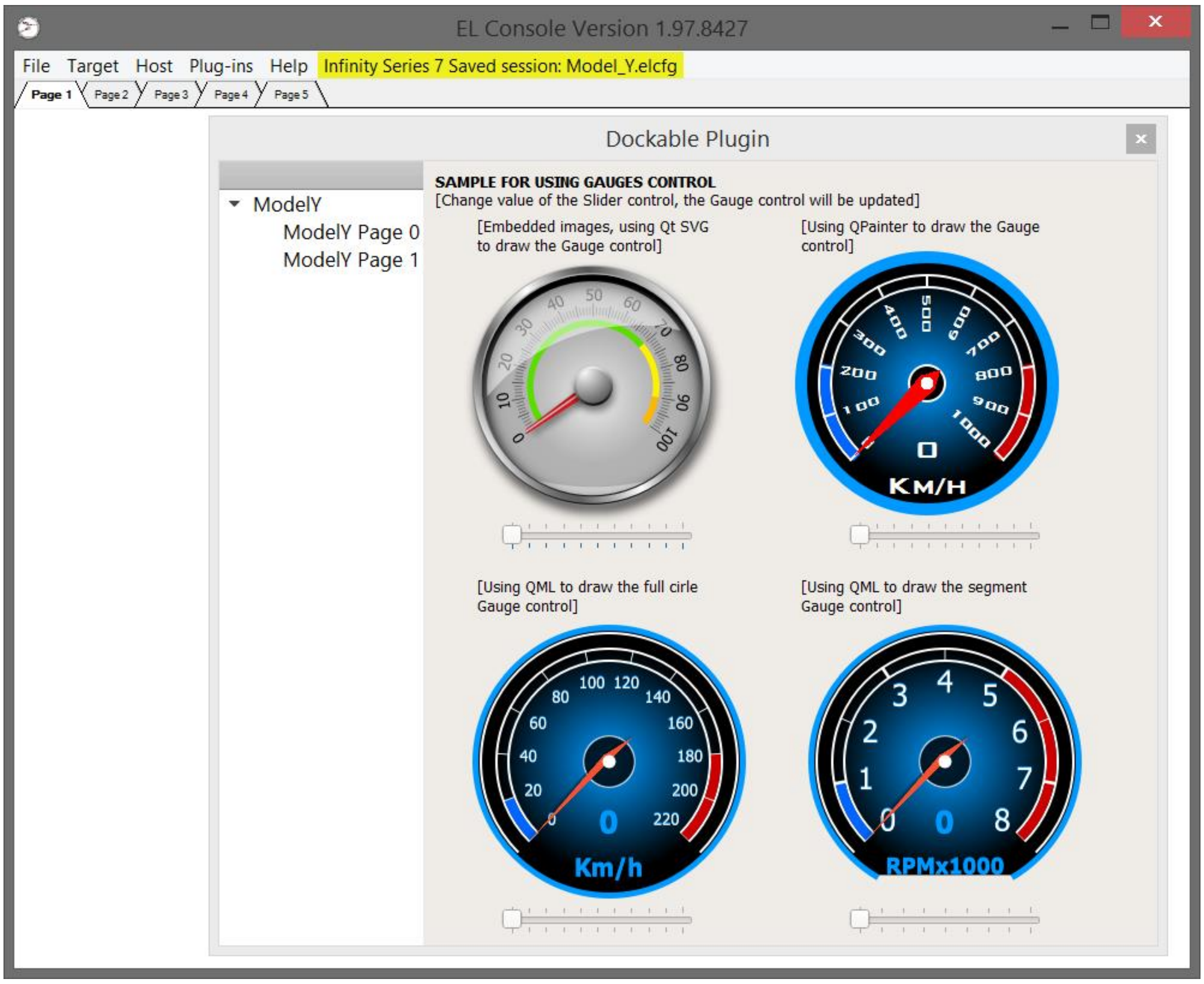

EL Console Version 1.97.8427
File
Target
Host
Plug-ins
Help
Infinity Series 7 Saved session: Model_Y.elcfg
Page 1
Page 2
Page 3
Page 4
Page 5
Dockable Plugin
ModelY
ModelY Page 0
ModelY Page 1
SAMPLE FOR USING GAUGES CONTROL
[Change value of the Slider control, the Gauge control will be updated]
[Embedded images, using Qt SVG to draw the Gauge control]
[Using QPainter to draw the Gauge control]
KM/H
[Using QML to draw the full cirle Gauge control]
Km/h
[Using QML to draw the segment Gauge control]
RPMx1000

Figure 32- The Plugin

4.3.1 Testing for ModelY Page 0

For more details, refer to

$(DocumentPath)\ EngineLab\EL Console\Plugins\EL_PythonQt\ModelY\ModelY.xml

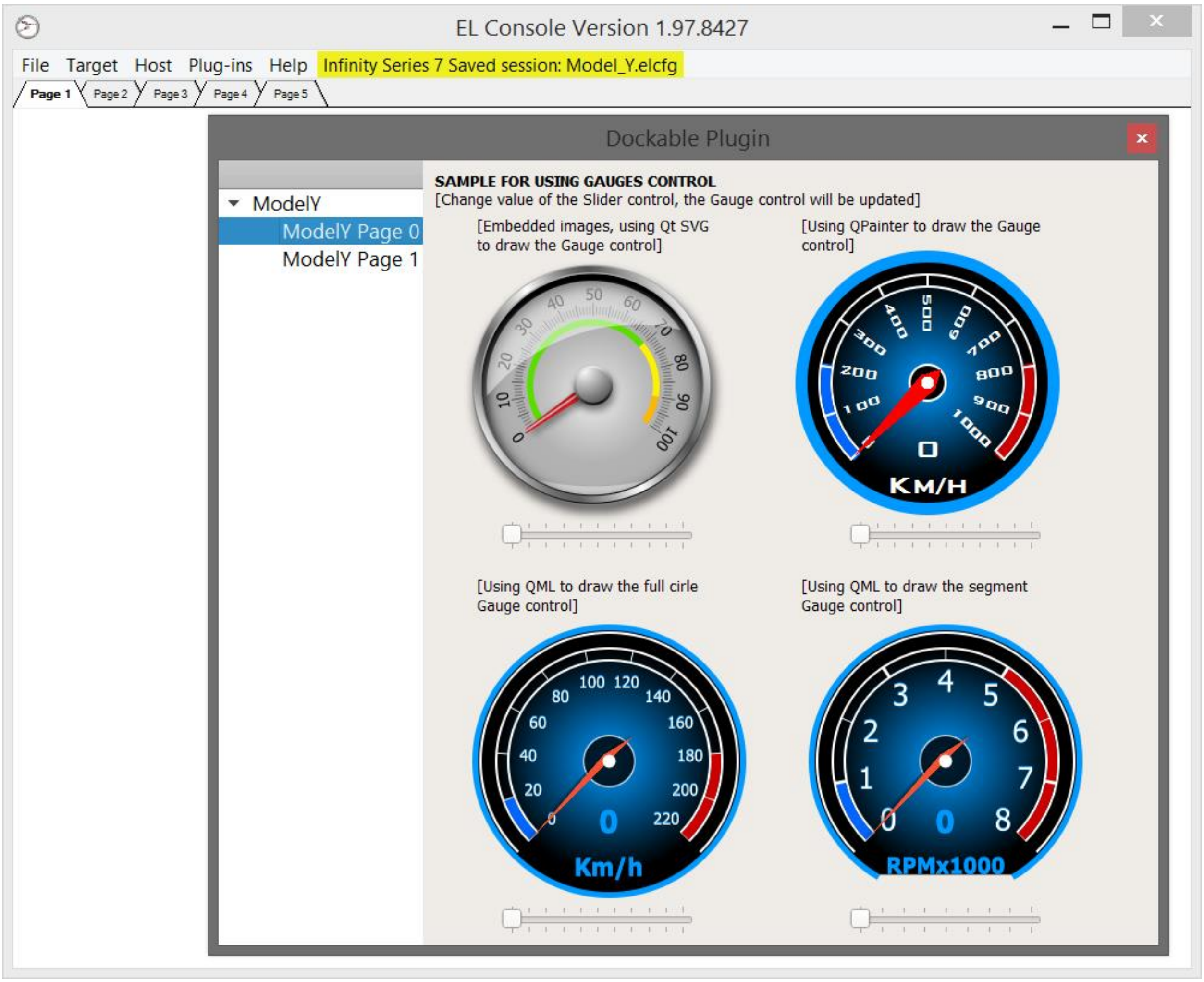
$(DocumentPath)\ EngineLab\EL Console\Plugins\EL_PythonQt\ModelY\ModelY_Page0.xml

$(DocumentPath)\ EngineLab\EL Console\Plugins\EL_PythonQt\ModelY\scripts\Py_ModelY_Page0.py

$(DocumentPath)\EngineLab\EL Console\Plugins\ EL_PythonQt \qml\main.qml

- ❖ **Note:** There are three types of Gauge. Refer to 5.1.4 Table controls for more details.

a) Click on ModelY Page 0 as picture below.



SAMPLE FOR USING GAUGES CONTROL
[Change value of the Slider control, the Gauge control will be updated]

[Embedded images, using Qt SVG to draw the Gauge control]

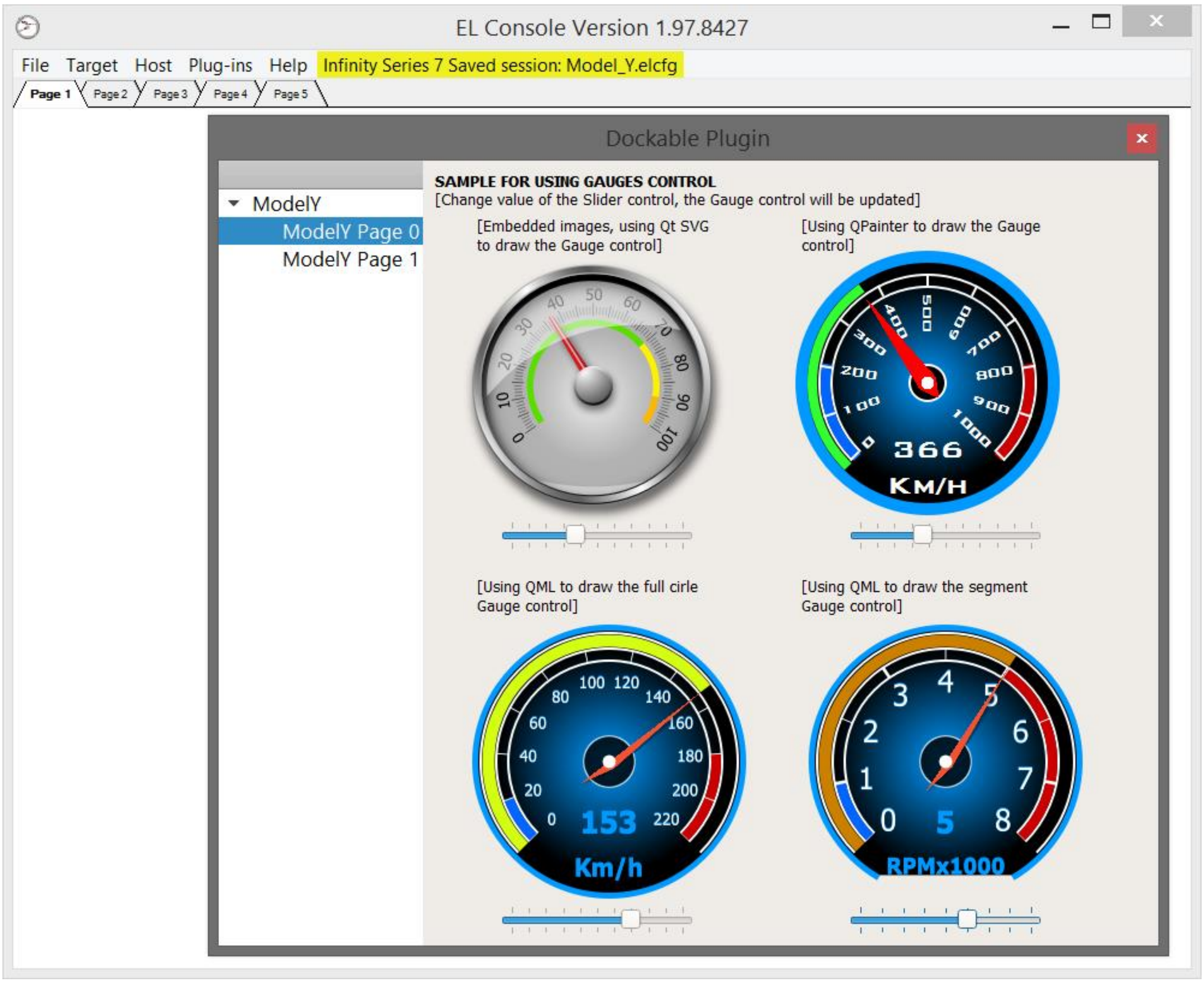
[Using QPainter to draw the Gauge control]

[Using QML to draw the full cirle Gauge control]

[Using QML to draw the segment Gauge control]

Figure 33-Loading ModelY Page 0

b) Pull slider to change value of EL Console channel and therefore the gauge will show the value of that channel.



SAMPLE FOR USING GAUGES CONTROL
[Change value of the Slider control, the Gauge control will be updated]

[Embedded images, using Qt SVG to draw the Gauge control]

[Using QPainter to draw the Gauge control]

[Using QML to draw the full cirle Gauge control]

[Using QML to draw the segment Gauge control]

Figure 34-Loading ModelY Page 0

4.3.2 Testing for ModelY Page 1

For more details, refer to

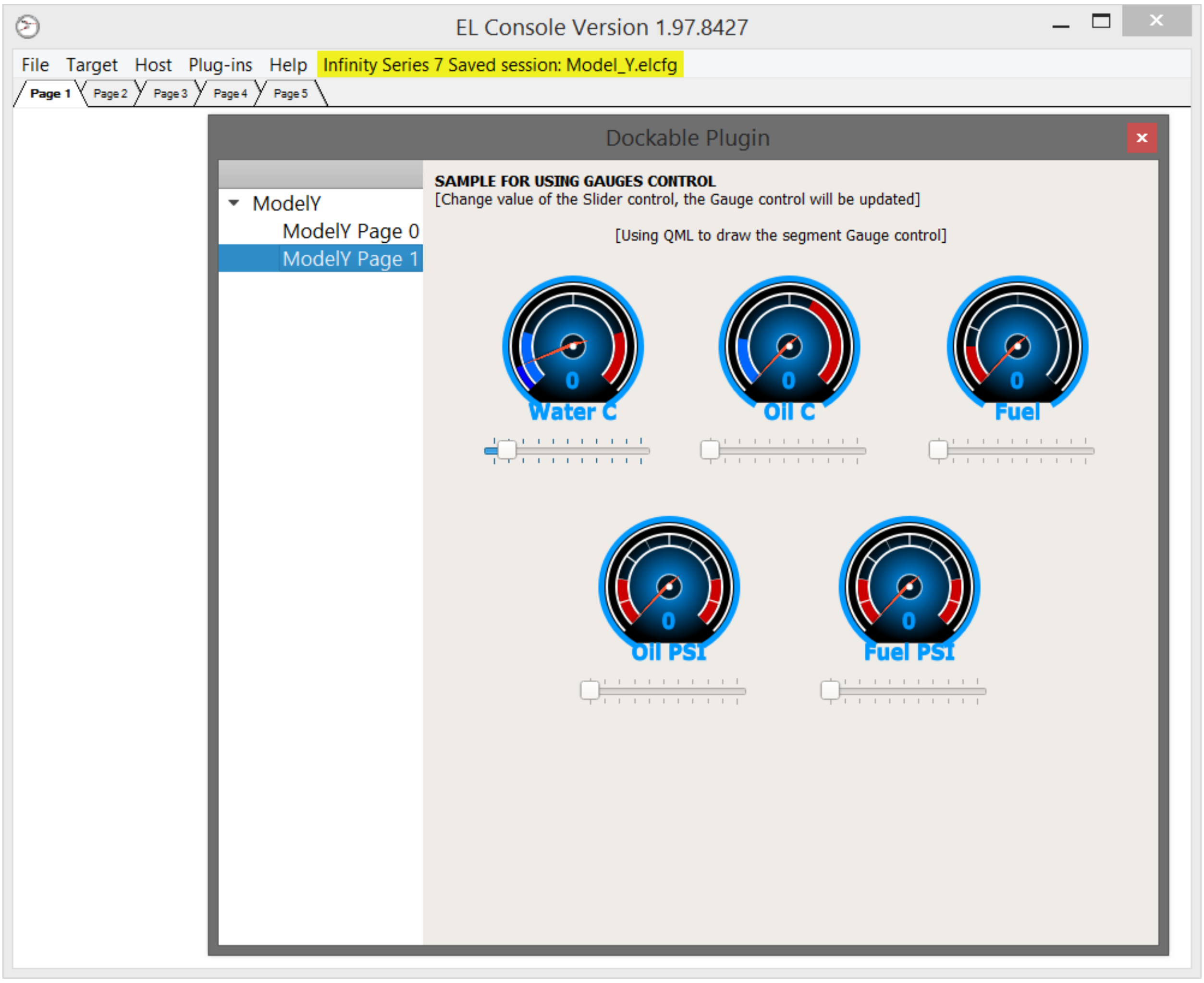
$(DocumentPath)\ EngineLab\EL Console\Plugins\EL_PythonQt\ModelY\ModelY.xml

$(DocumentPath)\ EngineLab\EL Console\Plugins\EL_PythonQt\ModelY\ModelY_Page1.xml

$(DocumentPath)\ EngineLab\EL Console\Plugins\EL_PythonQt\ModelY\scripts\Py_ModelY_Page1.py

$(DocumentPath)\EngineLab\EL Console\Plugins\ EL_PythonQt \qml\main.qml

a) Click on ModelY Page 1 as picture below.

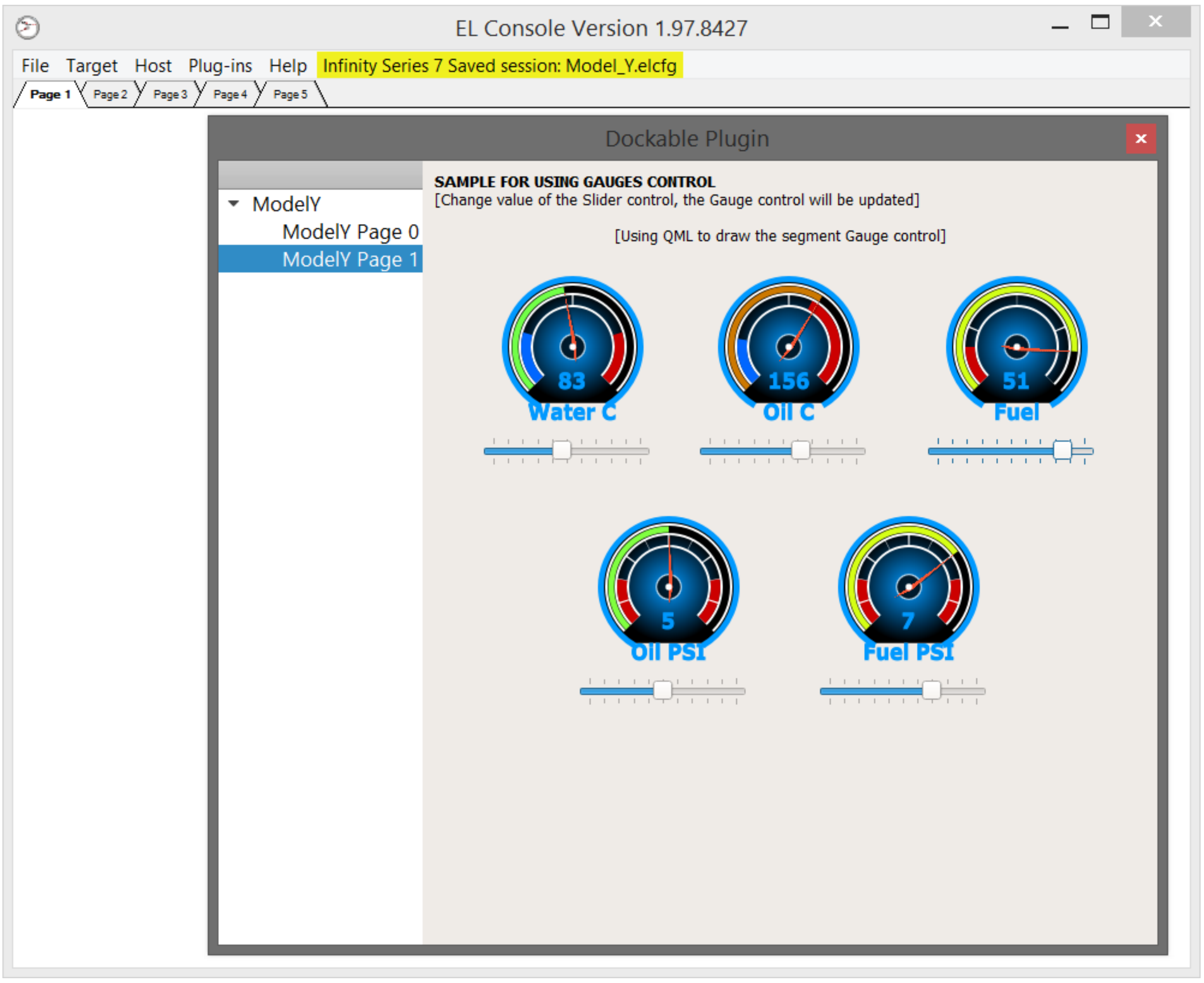


SAMPLE FOR USING GAUGES CONTROL
[Change value of the Slider control, the Gauge control will be updated]

[Using QML to draw the segment Gauge control]

Figure 35-Loading ModelY Page 1

b) Pull slider to change value of EL Console channel and therefore the gauge will show the value of that channel.



SAMPLE FOR USING GAUGES CONTROL
[Change value of the Slider control, the Gauge control will be updated]

[Using QML to draw the segment Gauge control]

Figure 36-Loading ModelY Page 1

5. How to use XML script file

You can use a text editor such as Notepad++ to open XML resource files below:

$(DocumentPath)\ EngineLab\EL Console\Plugins\EL_PythonQt\ModelX\ModelX.xml

$(DocumentPath)\ EngineLab\EL Console\Plugins\EL_PythonQt\ModelX\ModelX_Page0.xml

$(DocumentPath)\ EngineLab\EL Console\Plugins\EL_PythonQt\ModelX\ModelX_Page1.xml

$(DocumentPath)\ EngineLab\EL Console\Plugins\EL_PythonQt\ModelX\ModelX_Page2.xml

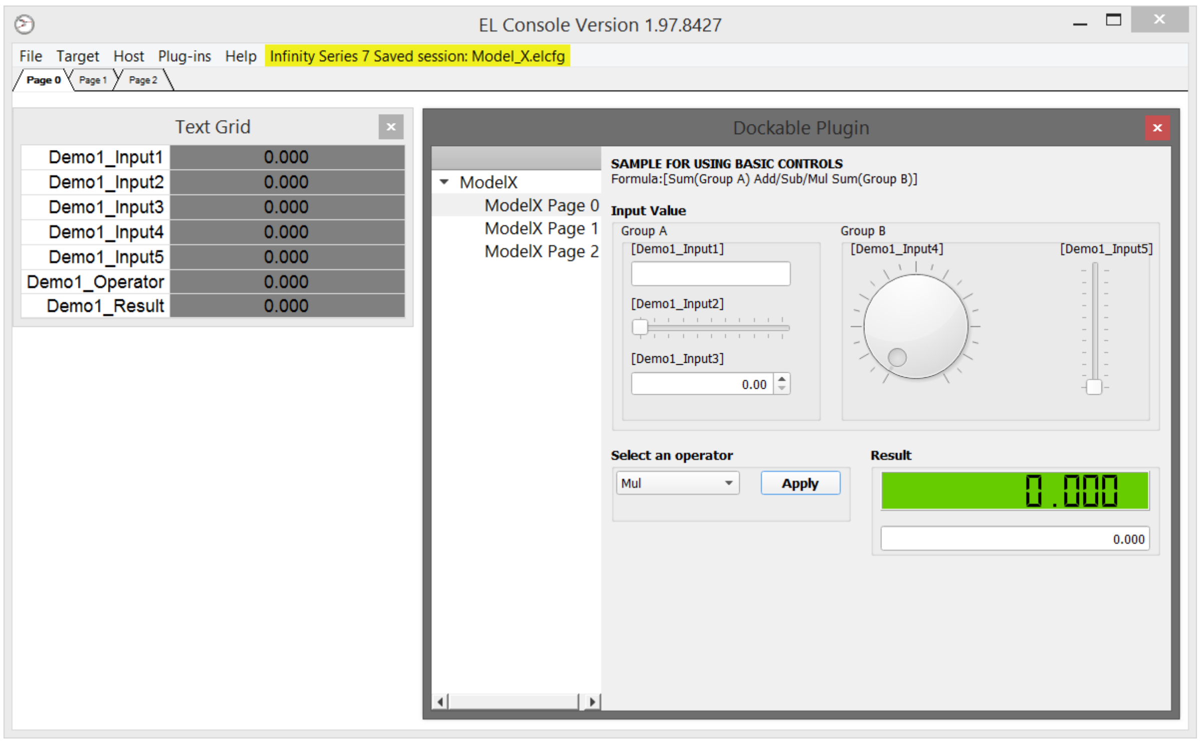
$(DocumentPath)\ EngineLab\EL Console\Plugins\EL_PythonQt\ModelY\ModelY.xml

$(DocumentPath)\ EngineLab\EL Console\Plugins\EL_PythonQt\ModelY\ModelY_Page0.xml

$(DocumentPath)\ EngineLab\EL Console\Plugins\EL_PythonQt\ModelY\ModelY_Page1.xml

$(DocumentPath)\ EngineLab\EL Console\Plugins\EL_PythonQt\ModelY\ModelY_Page2.xml

From resource XML file, when the plugin is loaded, the screen will be displayed as below.



Demo1_Input1	0.000
Demo1_Input2	0.000
Demo1_Input3	0.000
Demo1_Input4	0.000
Demo1_Input5	0.000
Demo1_Operator	0.000
Demo1_Result	0.000

SAMPLE FOR USING BASIC CONTROLS
Formula:[Sum(Group A) Add/Sub/Mul Sum(Group B)]

[Demo1_Input3] 0.00

Result 0.000 0.000

Figure 37-Main window of the Plugin

5.1 How to using XML file

We organize content of resource file (XML format) as below:

```
<plugin>
  <section name="ModelX">
        <page name="ModelX Page 0">
               <attr name=" filename" value="ModelX_Page0.xml" />
        </page>
        ...
  </section>
...
</plugin>
```

First we need to define the section of pages. Next are all pages in each section with the name of xml file which will contain the xml defined of each page. Then define separate page with its xml defined and put it into correct file name has been defined before. For example: Section "Model X" contains page "ModelX Page 0" and its xml defined of page "ModelX Page 0" will be place in file "ModelX_Page0.xml".

```
<page>
       <control type="text" rect="10 5 400 24">
              <attr name="caption" value="SAMPLE…" />
              ...
       </control>
</page>
```

XML resource file is designed base on Hierarchy Theory, with < plugin > block represent for main window of Plugin. In <plugin> block, we see that it has more <section> blocks. In <section> block, there are more <page> blocks. In <page> block, there are more <control > blocks. In <control> block, there are more <attr> blocks.

How to define parameter for each block type:

5.1.1 Section block

a) Format

```
<section name="Name of Section">
                ...
</section>
```

b) Usage

Define parameter for menu window. It is name of corresponding menu on it.

c) Example

Define name for the items in menu, with their name is “Section 1”.

```
<plugin>
    <section name="ModelX">
        <page name="Model X Page 0">
            <attr name=" filename" value="ModelX_Page0.xml" />
            …
        </page>
        …
    </section>
    …
</plugin>
```

You can add more < **section** > blocks in < **plugin** > block.

5.1.2 Page block

a) Format

```
<page>
        ...
</page>
```

b) Usage

Define parameter for page. It is name of corresponding page on it.

c) Example

Define name for the items in page, with their name is “Page1”

```
<page >
        <control type="text" rect="10 5 400 24">
                <attr name="caption" value="SAMPLE…" />
                …
        </control>
        …
</page>
```

You can add more < control > blocks in < page> block.

5.1.3 Control block

a) Format

```
<control type="type of the controls" rect="x y width height">
        ...
</control>
```

b) Usage

Define parameter for each window control of a window page, include two basic parameters, which is control type, rect(x, y, width, height).

About rect parameter, this is relative coordinate and size, compare with original coordinate – upper left corner of page window.

About control type parameter, value of this parameter must be referred from first column of Table in the section 7.1.4.

c) Example

Define the controls for a Page, with their control type is “text”, “dial”.

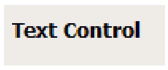
```
<page >
        <control type="text" rect="10 5 400 24">
                <attr name="caption" value="SAMPLE…" />
                …
        </control>
        <control type="text" rect="10 20 400 24">
                <attr name="caption" value="Formula: …" />
                …
        </control>
        …
</page>
```

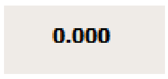
You can add more <attr > blocks in <control> block.

You already understand how to use XML resource file to custom UI and set value for channel variable, now it's time to build a new page on your own.

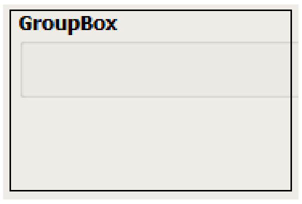
5.1.4 Table controls

TYPE	Example code	SAMPLE IMAGE
Text Control		

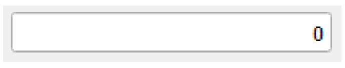
Text Input	`<control type="text" rect="10 5 400 24">`
	`<attr name="id" value="1" />`
	`<attr name="caption" value="Text Control" />`
	`<attr name="style" value="bold" />`
	`</control>`
		 Text Control

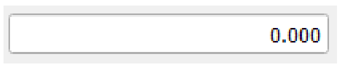
Text Output	`<control type="text" rect="10 5 400 24">`
	`<attr name="id" value="1" />`
	`<attr name="style" value="bold" />`
	`<attr name="read_channel" value="Demo1_Input1" />`
	`</control>`
		 0.000

Groupbox Control		

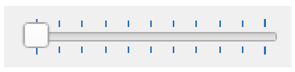
Groupbox	`<control type="groupbox" rect="10 300 240 56">`
	`<attr name="id" value="1" />`
	`<attr name="caption" value="GroupBox" />`
	`<attr name="style" value="bold" />`
	`</control>`
		 GroupBox

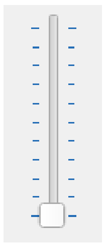
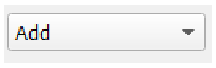
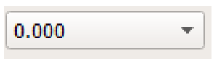
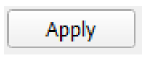
Editbox Control		

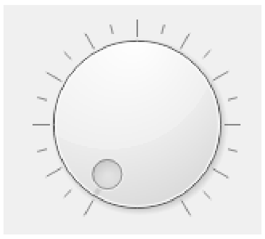
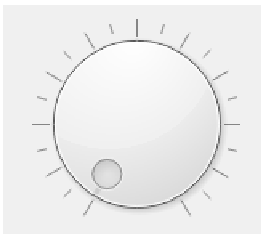
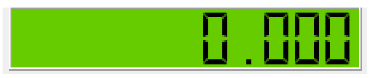
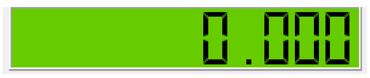
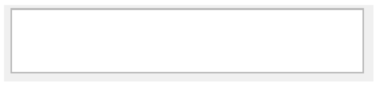
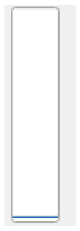
Editbox Input	`<control type="editbox" rect="30 115 200 25">`
	`<attr name="id" value="1" />`
	`<attr name="textChanged_event" value="demo1 OnEdit" />`
	`</control>`
		 0

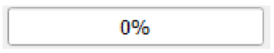
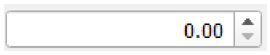
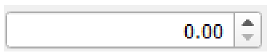
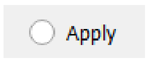
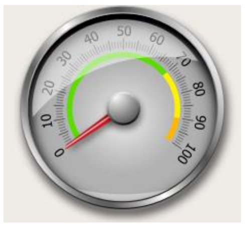
Editbox Output	`<control type="editbox" rect="30 115 200 25">`
	`<attr name="id" value="1" />`
	`<attr name="read_channel" value="Demo1_Result" />`
	`</control>`
	`<attr name="textChanged_event" value="demo1 OnEdit" />`
		 0.000

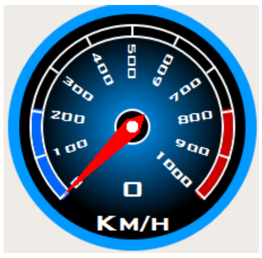
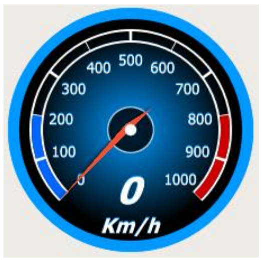
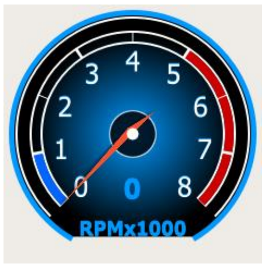
Slider Control		

Slider Input	`<control type="slider " rect="30 170 200 25">`
	`<attr name="style" value="horizontal" />`
		

	<attr name="id" value="8" /> <attr name="range" value="0 100 1" /> <attr name="valueChanged_event" value="demo1 OnSliderHorizontal" /> </control>	
Slider Output	<control type="slider " rect="480 115 30 135"> <attr name="style" value="vertical" /> <attr name="id" value="8" /> <attr name="range" value="0 100 1" /> <attr name="read_channel" value="Demo1_Input2" /> <attr name="valueChanged_event" value="demo1 OnSliderVertical" /> </control>	
Combobox Control		
Combobox Input	<control type="combobox" rect="15 325 124 24"> <attr name="id" value="16" /> <attr name="item" value="Add" /> <attr name="item" value="Sub" /> <attr name="item" value="Mul" /> <!--<attr name="highlighted_event" value="demo1 test" />--> </control>	
Combobox Output	<control type="combobox" rect="15 325 124 24"> <attr name="id" value="16" /> <attr name="item" value="" /> <attr name="read_channel" value="Demo1_Operator" /> <!--<attr name="highlighted_event" value="demo1 test" />--> </control>	
Button Control		
Button	<control type="button" rect="160 325 80 24"> <attr name="id" value="17" /> <attr name="caption" value="Apply" /> <attr name="clicked_event" value="demo1 apply" /> </control>	
Dial Control		

Dial Input	<control type="dial" rect="250 115 130 130"> <attr name="id" value="13" /> <attr name="range" value="0 100 1" /> <attr name="valueChanged_event" value="demo1 OnDial" /> </control>	
Dial Output	<control type="dial" rect="250 115 130 130"> <attr name="id" value="13" /> <attr name="range" value="0 100 1" /> <attr name="read_channel" value="Demo1_Input4" /> <attr name="valueChanged_event" value="demo1 OnDial" /> </control>	
LCD Control		
LCD input	<control type="lcd" rect="20 435 220 40"> <attr name="id" value="19" /> </control>	
LCD output	<control type="lcd" rect="20 435 220 40"> <attr name="id" value="19" /> <attr name="read_channel" value="Demo1_Result" /> </control>	
Table Control		
Table	<control type="table" rect="20 435 220 40"> <attr name="id" value="19" /> <attr name="cellClicked_event" value="demo1 test" /> </control>	
ProgressBar Control		
ProgressBar Input	<control type="progressbar" rect="480 115 30 135"> <attr name="style" value="vertical" /> <attr name="id" value="8" /> <attr name="range" value="0 100 1" /> <attr name="valueChanged_event" value="demo1 OnProgressbar" /> </control>	

ProgressBar Output	<control type="progressbar " rect="30 170 200 25"> <attr name="style" value="horizontal" /> <attr name="id" value="8" /> <attr name="range" value="0 100 1" /> <attr name="read_channel" value="Demo1_Input2" /> <attr name="valueChanged_event" value="demo1 OnProgressbar" /> </control>	 0%
Spinbox Control		
Spinbox Input	<control type="spinbox" rect="30 225 160 25"> <attr name="id" value="10" /> <attr name="range" value="0 100 1" /> <attr name="valueChanged_event" value="demo1 OnSpinbox" /> </control>	 0.00
Spinbox Output	<control type="spinbox" rect="30 225 160 25"> <attr name="id" value="10" /> <attr name="range" value="0 100 1" /> <attr name="read_channel" value="Demo1_Input3" /> <attr name="valueChanged_event" value="demo1 OnSpinbox" /> </control>	 0.00
RadioButton Control		
RadioButton Control	<control type="radiobutton" rect="160 325 80 24"> <attr name="id" value="17" /> <attr name="caption" value="Apply" /> <attr name="clicked_event" value="demo1 apply" /> </control>	 Apply
Gauge Control		
Gauge Control (Using Qt SVG render)	<control type="gauge" rect="30 80 230 230"> <attr name="id" value="55" /> <attr name="range" value="0 100" /> <attr name="read_channel" value="Gauge_Test1" /> </control>	 0 10 20 30 40 50 60 70 80 90 100

Gauge Control (Using QPainter)	<control type="cargauge" rect="305 70 230 -1"> <attr name="id" value="57" /> <attr name="caption" value="Km/h" /> <attr name="range" value="0 1000 200 800" /> <attr name="read_channel" value="Gauge_Test2" /> </control>	 100 200 300 400 500 600 700 800 900 1000 0 KM/H
Gauge Control (Using QML)	<control type="qmlcargauge" rect="25 380 250 250"> <attr name="style" value="full" /> <attr name="id" value="60" /> <attr name="caption" value="Km/h" /> <attr name="range" value="0 1000 200 800" /> <attr name="read_channel" value="Gauge_Test3" /> </control>	 0 100 200 300 400 500 600 700 800 900 1000 0 Km/h
Gauge Control (Using QML)	<control type="qmlcargauge" rect="305 370 260 260"> <attr name="style" value="segment" /> <attr name="id" value="63" /> <attr name="color" value="#0066FF #cc0000" /> <attr name="caption" value="RPMx1000" /> <attr name="text_size" value="15 15 11" /> <attr name="range" value="0 8 1 5 7 10" /> <attr name="read_channel" value="Gauge_Test4" /> </control>	

b) **Note:** If you want to access and configure a control, you will have to define attribute “id” to that control. Later you will find that control through its id and use suitable function to configure it.

5.2 Change name of control in XML file

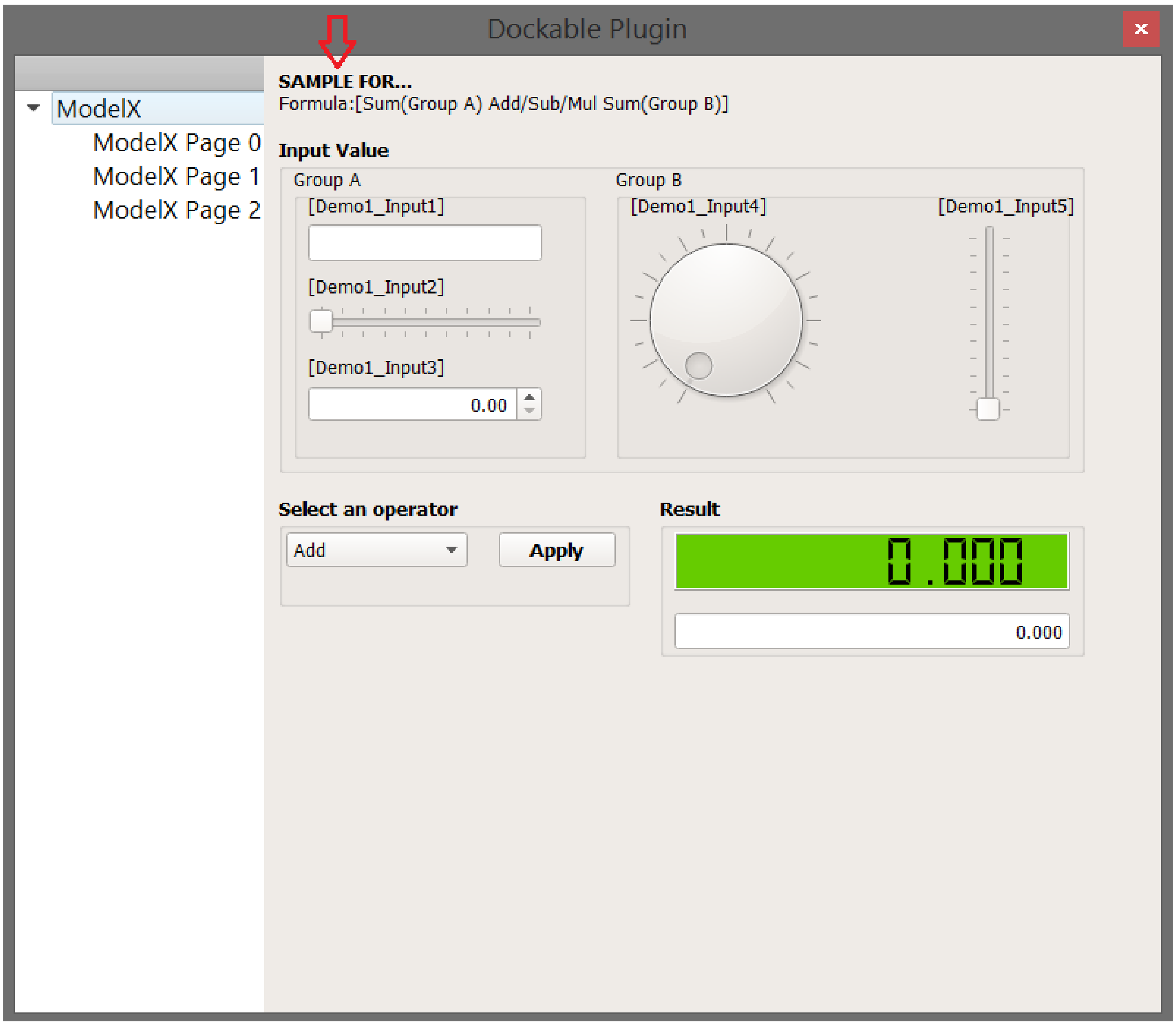
Suppose we change name of the lable control, you follow the steps as below

a) You find the first control in XML file as image below.

```
<control type="text" rect="10 5 400 24">
        <attr name="caption" value="SAMPLE FOR..." />
        <attr name="style" value="bold" />
 </control>
```

b) You replace text: “**SAMPLE FOR USING BASIC CONTROLS**” to “**SAMPLE FOR ...**”

c) Save the file and reload the Plug-in, the screen is display as below.



Dockable Plugin

ModelX
ModelX Page 0
ModelX Page 1
ModelX Page 2

SAMPLE FOR...
Formula:[Sum(Group A) Add/Sub/Mul Sum(Group B)]

Input Value

Group A
[Demo1_Input1]
[Demo1_Input2]
[Demo1_Input3] 0.00

Group B
[Demo1_Input4]
[Demo1_Input5]

Select an operator
Add
Apply

Result
0.000
0.000

Figure 38-Change name of control

5.3 Change the control by other control

Suppose we replace the control of [Demo1_Input1], you follow the steps as below.

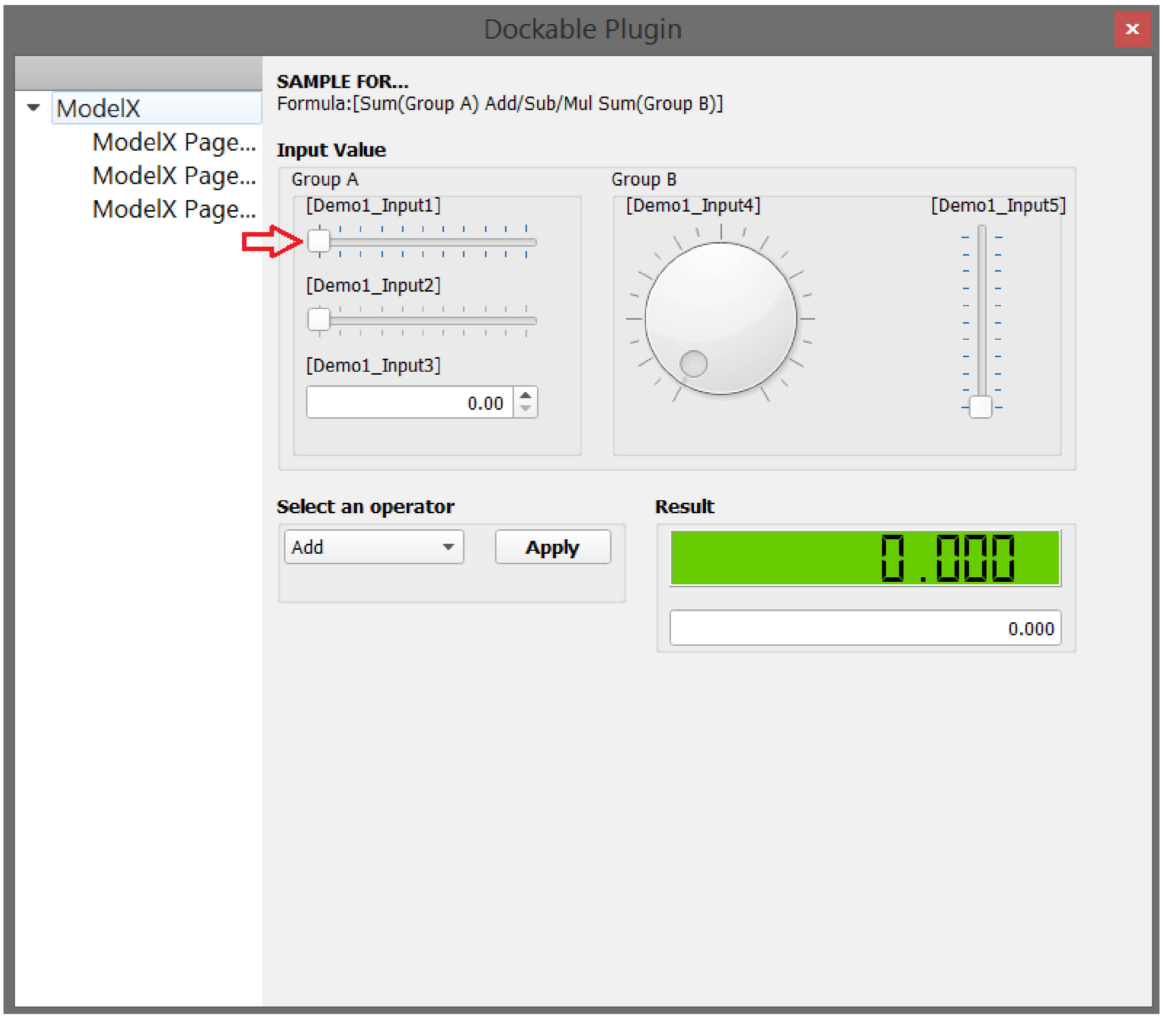
a) You find code in .xml file as below.

```
<control type="edit" rect="30 115 160 25">
        <attr name="id" value="6" />
        <attr name="read_channel" value="Demo1_Input1" />
</control>
```

b) You replace code above by code as below.

```
<control type="slider" rect="30 115 160 25">
        <attr name="id" value="6" />
        <attr name="style" value="horizontal" />
        <attr name="read_channel" value="Demo1_Input1" />
</control>
```

c) Save the file and reload the Plug-in, the screen will be displayed as below.



SAMPLE FOR...
Formula:[Sum(Group A) Add/Sub/Mul Sum(Group B)]

[Demo1_Input3] 0.00

Result 0.000

0.000

Figure 39-Replace control

5.4 Change name of section

Suppose we change name of section, you follow the steps as below.

a) You find code in .xml file as below.

```
<section name="ModelX">
```

b) You replace code above by code as below.

```
<section name="Section 1">
```

c) Save the file and reload the Plug-in, the screen will be displayed as below.

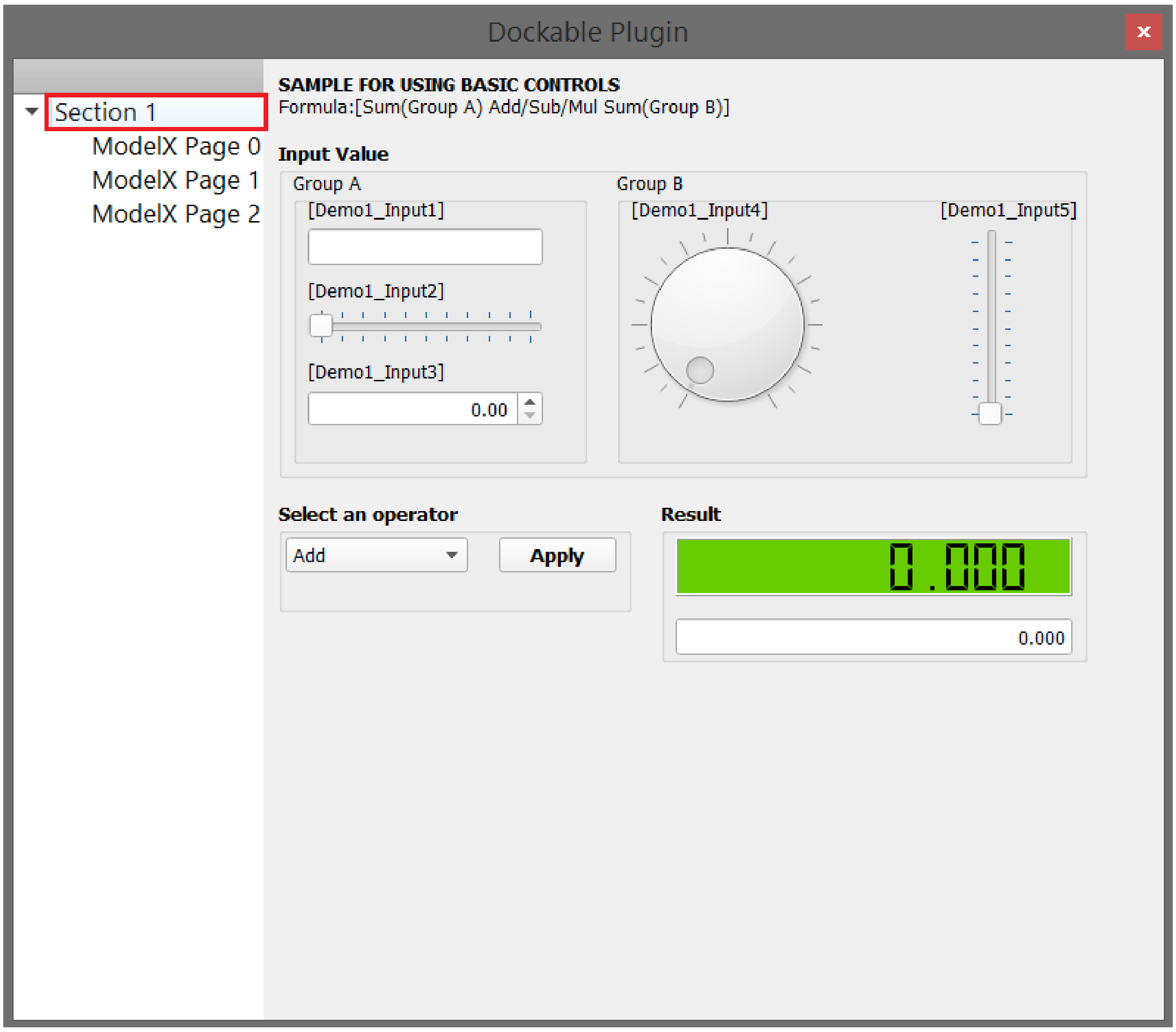


Figure 40-Change name of session

5.5 Change name of page

Suppose we change name of Page, you follow the steps as below.

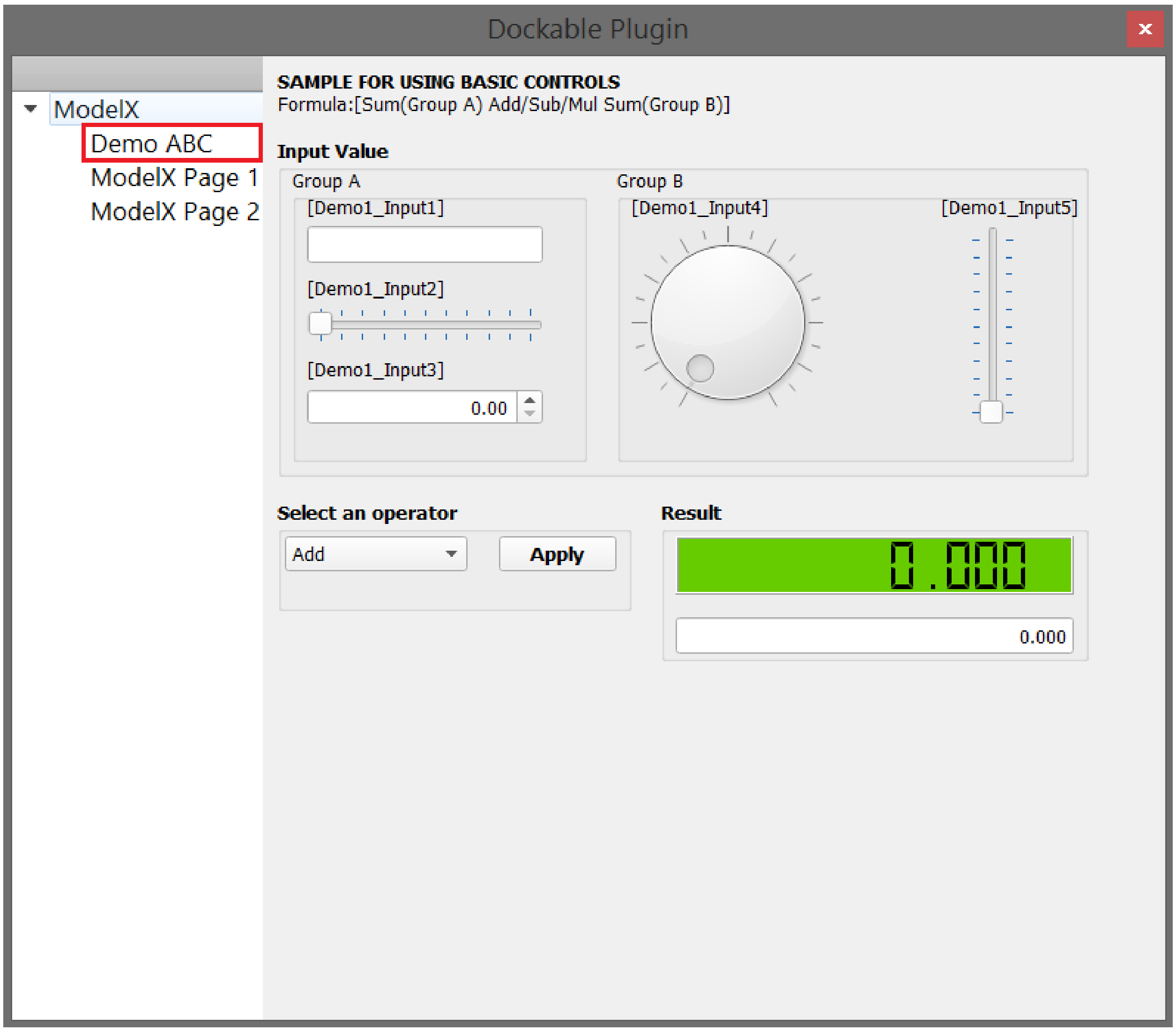
a) You find code in .xml file (find by page, each page has its xml defined code) as below.

```
<page name="ModelX Page 0">
```

b) You replace code above by code as below.

```
<page name="Demo ABC">
```

c) Save the file and reload the Plug-in, the screen will be displayed as below.



SAMPLE FOR USING BASIC CONTROLS
Formula:[Sum(Group A) Add/Sub/Mul Sum(Group B)]

Figure 41-Change name of Page

5.6 Add a section to main windows

Suppose we add section to main windows, you follow the steps as below

a) Define a new section as below. This xml code shows that section “Section” has a page, this page xml defined code will be place in file” ModelX.xml”.

```
<section name="Section">
        <page name="Page 1">
                <attr name="filename" value="ModelX_Page0.xml" />
        </page>
</section>
```

b) Then open file “ModelX_Page0.xml” and define attribute of page “Page 1”

```
<page >
        <control type="text" rect="20 20 300 24">
                <attr name="id" value="37" />
                <attr name="caption" value="SAMPLE FOR MODIFYING TABLE CELL 1D/2D" />
                <attr name="style" value="bold" />
        </control>
        <control type="text" rect="20 35 300 24">
                <attr name="id" value="38" />
                <attr name="caption" value="(Set values for Table cell)" />
        </control>
        …
</page>
```

c) Save the file and reload the Plug-in, the screen will be displayed as below.

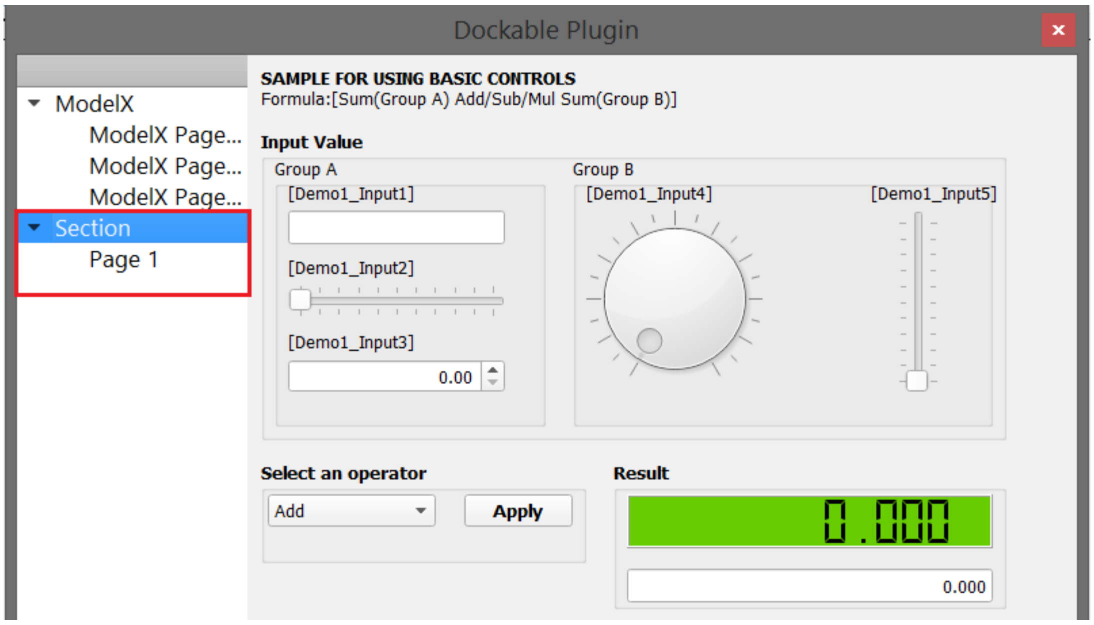


Figure 42-Add section to main window

6. How to use Python script file

6.1 How to modify Python code

Make sure you already had all necessary files for the plugin, then start the EL Console and load the Plugin as described in the Section 4.

You can use a text editor such as Notepad++ to open XML resource files below:

$(DocumentPath)\ EngineLab\EL Console\Plugins\EL_PythonQt\ModelX\ModelX.xml

$(DocumentPath)\ EngineLab\EL Console\Plugins\EL_PythonQt\ModelX\ModelX_Page0.xml

$(DocumentPath)\ EngineLab\EL Console\Plugins\EL_PythonQt\ModelX\ModelX_Page1.xml

$(DocumentPath)\ EngineLab\EL Console\Plugins\EL_PythonQt\ModelX\ModelX_Page2.xml

$(DocumentPath)\EngineLab\EL Console\Plugins\ EL_PythonQt\ModelX \scripts\common.py

$(DocumentPath)\EngineLab\EL Console\Plugins\ EL_PythonQt\ModelX \scripts\init.py

$(DocumentPath)\EngineLab\EL Console\Plugins\ EL_PythonQt\ModelX \scripts\Py_ModelX_Page0.py

$(DocumentPath)\EngineLab\EL Console\Plugins\ EL_PythonQt\ModelX \scripts\Py_ModelX_Page1.py

$(DocumentPath)\EngineLab\EL Console\Plugins\ EL_PythonQt\ModelX \scripts\Py_ModelX_Page2.py

$(DocumentPath)\EngineLab\EL Console\Plugins\ EL_PythonQt\ModelX \scripts\PyScriptTestQtControl.py

$(DocumentPath)\ EngineLab\EL Console\Plugins\EL_PythonQt\ModelY\ModelY.xml

$(DocumentPath)\ EngineLab\EL Console\Plugins\EL_PythonQt\ModelY\ModelY_Page0.xml

$(DocumentPath)\ EngineLab\EL Console\Plugins\EL_PythonQt\ModelY\ModelY_Page1.xml

$(DocumentPath)\ EngineLab\EL Console\Plugins\EL_PythonQt\ModelY\ModelY_Page2.xml

$(DocumentPath)\EngineLab\EL Console\Plugins\ EL_PythonQt\ModelY \scripts\common.py

$(DocumentPath)\EngineLab\EL Console\Plugins\ EL_PythonQt\ ModelY \scripts\init.py

$(DocumentPath)\EngineLab\EL Console\Plugins\ EL_PythonQt\ ModelY \scripts\Py_ ModelY _Page0.py

$(DocumentPath)\EngineLab\EL Console\Plugins\ EL_PythonQt\ ModelY \scripts\Py_ ModelY _Page1.py

$(DocumentPath)\EngineLab\EL Console\Plugins\ EL_PythonQt\ModelY \scripts\ PyScriptTestQtControl.py

6.1.1 The functions for the Python script

Look into source code in **common.py** file, this supports 5 primary functions and 14 Qt control classes.

- **printfString(text):** **(1)**
- **EL_CreateNoOpChannel** **(2)**
- **EL_WriteNoOpChannel(name, value):** **(3)**
- **EL_NoOpChannelValue(name):** **(4)**
- **EL_WriteTableChannel(name, info_select, x_axis, value):** **(5)**
- **EL_TableChannelValue(name, info_select, x_axis):** **(6)**
- **class CommonControl:** **(7)**
- **class EditControl(CommonControl):** **(8)**
- **class RichEditControl(CommonControl):** **(9)**
- **class ComboboxControl(CommonControl):** **(10)**
- **class SliderControl(CommonControl):** **(11)**
- **class SpinBoxControl(CommonControl):** **(12)**
- **class TextControl(CommonControl):** **(13)**
- **class CheckboxControl(CommonControl):** **(14)**
- **class ButtonControl(CommonControl):** **(15)**
- **class DialControl(CommonControl):** **(16)**
- **class LCDControl(CommonControl):** **(17)**
- **class ProgressbarControl(CommonControl):** **(18)**
- **class QTableControl(CommonControl):** **(19)**

FUNCTION OR CLASS	USAGE
(1)	Print string text to debugging dialog. Passed argument is text need to be display.
(2)	Create "NoOp" channel in current model. Passed arguments are name of the channel.
(3)	Set a value of a "NoOp" channel in current model. Passed arguments are name and value of the channel.
(4)	Return a value of a "NoOp" channel in current model. Passed arguments are name of the channel.
(5)	Set a value of a "Table" channel in current model. Passed arguments are name, info_select (type of value), x_asix (coordinate of value in the table) and value of the channel.

(6)	Return a value of a “Table” channel in current model. Passed arguments are name, info_select (type of value), x_asix (coordinate of value in the table) value of the channel.
(7)	Provide common functions to access and configure all kind of Qt control.
(8)	Inherit all common functions of class CommonControl and provide common functions to access and configure all Editbox Control.
(9)	Inherit all common functions of class CommonControl and provide common functions to access and configure all RichEditbox Control.
(10)	Inherit all common functions of class CommonControl and provide common functions to access and configure all Combobox Control.
(11)	Inherit all common functions of class CommonControl and provide common functions to access and configure all Slider Control.
(12)	Inherit all common functions of class CommonControl and provide common functions to access and configure all Spinbox Control.
(13)	Inherit all common functions of class CommonControl and provide common functions to access and configure all Text Control.
(14)	Inherit all common functions of class CommonControl and provide common functions to access and configure all Checkbox Control.
(15)	Inherit all common functions of class CommonControl and provide common functions to access and configure all Button Control.
(16)	Inherit all common functions of class CommonControl and provide common functions to access and configure all Dial Control.
(17)	Inherit all common functions of class CommonControl and provide common functions to access and configure all LCD Control.
(18)	Inherit all common functions of class CommonControl and provide common functions to access and configure all Progressbar Control.
(19)	Inherit all common functions of class CommonControl and provide common functions to access and configure all Table Control.

6.2 Example for modifying Python code in ModelX Page 0

Suppose you add a channel output "Demo1_Input2_Python" for edit box. You have to modify Python code and XML file; you follow the steps as below.

a) Open **Py_ModelX_Page0.py** file to edit, you can find this file at the location as below.

$(DocumentPath)\EngineLab\EL Console\Plugins\ EL_PythonQt\ModelX \scripts\Py_ModelX_Page0.py

Find lines of code in **Py_ModelX_Page0.py** file:

```
def OnSliderHorizontal():
    objSlider = SliderControl()
    EL_WriteNoOpChannel("Demo1_Input2", objSlider.value("8"))
```

b) Change text "**Demo1_Input2**" to "**Demo1_Input2_Python**" as image below

```
def OnSliderHorizontal():
    objSlider = SliderControl()
    EL_WriteNoOpChannel("Demo1_Input2_Python", objSlider.value("8"))
```

c) Open **ModelX_Page0.xml** file, you can find this file at the location as below.

$(DocumentPath)\ EngineLab\EL Console\Plugins\EL_PythonQt\ModelX\ModelX_Page0.xml

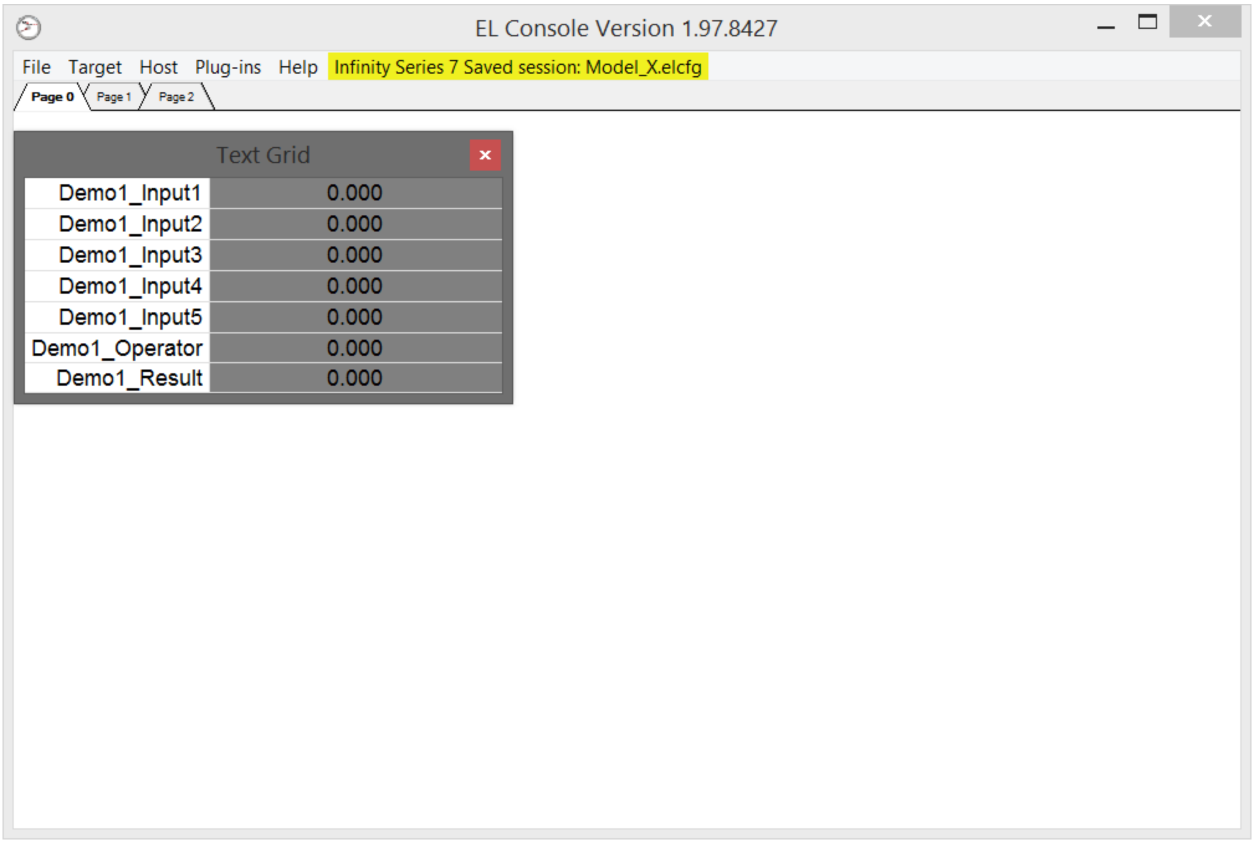
Find the lines of code in **ModelX_Page0.xml** file:

```
<control type="editbox" rect="30 115 160 25">
            <attr name="id" value="6" />
            <attr name="textChanged_event" value="Py_ModelX_Page0 OnEditbox" />
  </control>
```

d) Add text "**<attr name="read_channel" value="Demo1_Input2_Python" />**" as image below

```
<control type="editbox" rect="30 115 160 25">
            <attr name="id" value="6" />
            <attr name="read_channel" value="Demo1_Input2_Python" />
  </control>
```

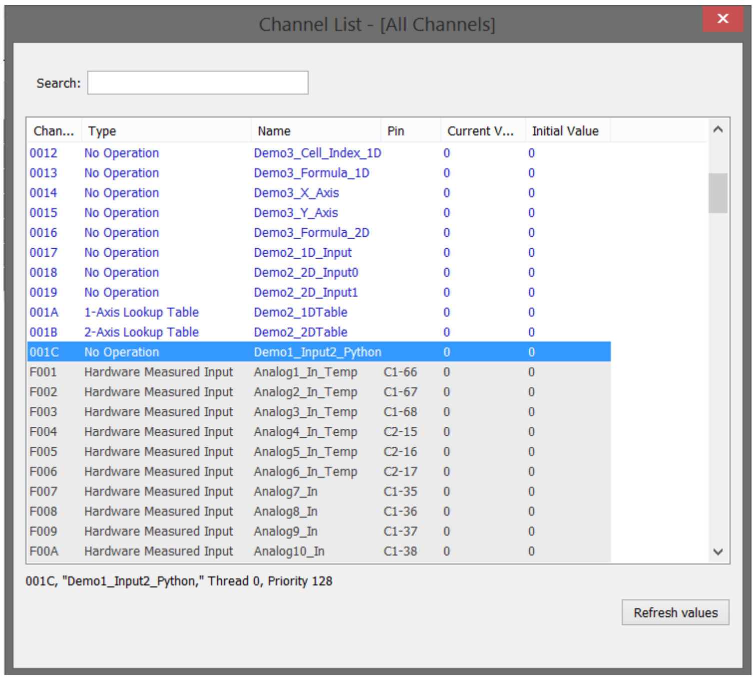
e) Save the files (.xml and .py). Start EL Console, and load session and layout file, the screen is displayed picture as below.


EL Console Version 1.97.8427
File Target Host Plug-ins Help
Infinity Series 7 Saved session: Model_X.elcfg
Page 0 Page 1 Page 2
Text Grid

Demo1_Input1	0.000
Demo1_Input2	0.000
Demo1_Input3	0.000
Demo1_Input4	0.000
Demo1_Input5	0.000
Demo1_Operator	0.000
Demo1_Result	0.000

Figure 43-The main window of the Plugin

f) Select Plug-ins-> PythonQt Plugin->Load Plugin. When defined a “read_channel” in xml file, if that channel did not exist it will be auto created. Select Targer -> Channel List, the screen is displayed as picture below.



Channel List - [All Channels]

Search:

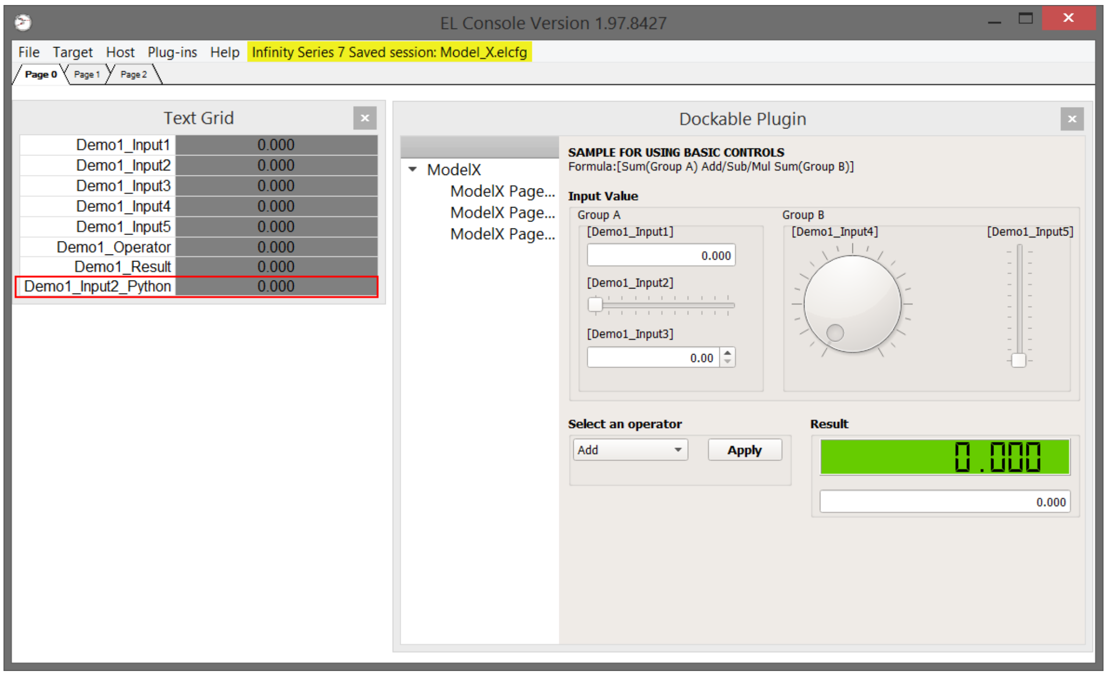
Chan...	Type	Name	Pin	Current V...	Initial Value
0012	No Operation	Demo3_Cell_Index_1D		0	0
0013	No Operation	Demo3_Formula_1D		0	0
0014	No Operation	Demo3_X_Axis		0	0
0015	No Operation	Demo3_Y_Axis		0	0
0016	No Operation	Demo3_Formula_2D		0	0
0017	No Operation	Demo2_1D_Input		0	0
0018	No Operation	Demo2_2D_Input0		0	0
0019	No Operation	Demo2_2D_Input1		0	0
001A	1-Axis Lookup Table	Demo2_1DTable		0	0
001B	2-Axis Lookup Table	Demo2_2DTable		0	0
001C	No Operation	Demo1_Input2_Python		0	0
F001	Hardware Measured Input	Analog1_In_Temp	C1-66	0	0
F002	Hardware Measured Input	Analog2_In_Temp	C1-67	0	0
F003	Hardware Measured Input	Analog3_In_Temp	C1-68	0	0
F004	Hardware Measured Input	Analog4_In_Temp	C2-15	0	0
F005	Hardware Measured Input	Analog5_In_Temp	C2-16	0	0
F006	Hardware Measured Input	Analog6_In_Temp	C2-17	0	0
F007	Hardware Measured Input	Analog7_In	C1-35	0	0
F008	Hardware Measured Input	Analog8_In	C1-36	0	0
F009	Hardware Measured Input	Analog9_In	C1-37	0	0
F00A	Hardware Measured Input	Analog10_In	C1-38	0	0

001C, "Demo1_Input2_Python," Thread 0, Priority 128

Refresh values

Figure 44-Channel list

g) Click right mouse on TextGrid control -> set channel ->find and check on “**Demo1_Input2_Python**” channel, the screen is displayed as picture below.



EL Console Version 1.97.8427

File Target Host Plug-ins Help Infinity Series 7 Saved session: Model_X.elcfg

Page 0 Page 1 Page 2

Text Grid

Demo1_Input1	0.000
Demo1_Input2	0.000
Demo1_Input3	0.000
Demo1_Input4	0.000
Demo1_Input5	0.000
Demo1_Operator	0.000
Demo1_Result	0.000
Demo1_Input2_Python	0.000

Dockable Plugin

ModelX
ModelX Page...
ModelX Page...
ModelX Page...

SAMPLE FOR USING BASIC CONTROLS
Formula:[Sum(Group A) Add/Sub/Mul Sum(Group B)]

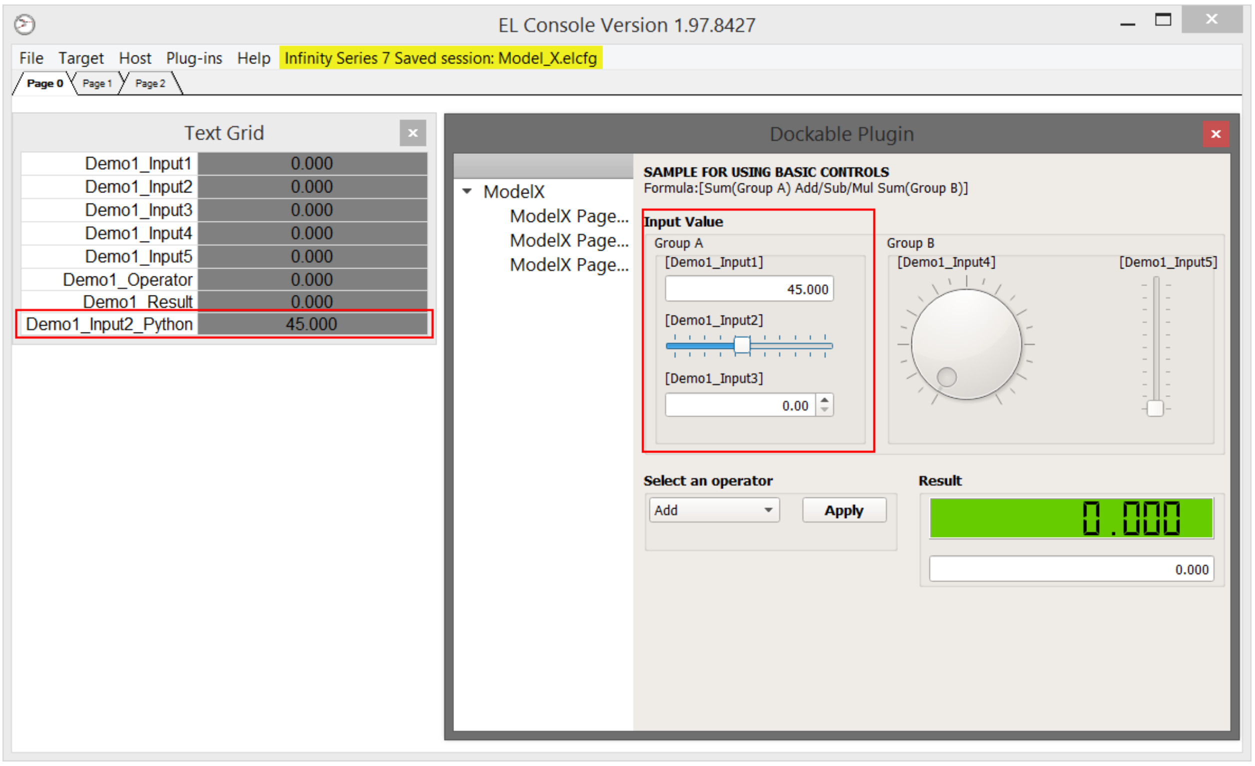
Input Value
Group A
[Demo1_Input1] 0.000
[Demo1_Input2]
[Demo1_Input3] 0.00
Group B
[Demo1_Input4]
[Demo1_Input5]

Select an operator: Add | Apply

Result: 0.000 | 0.000

Figure 45-Set channel for the control

h) Setting values of horizontal slider, the "**Demo1_Input2_Python**"channel and the edit control which define "read_channel" "**Demo1_Input2_Python**" will be updated value as picture below.



Demo1_Input1	0.000
Demo1_Input2	0.000
Demo1_Input3	0.000
Demo1_Input4	0.000
Demo1_Input5	0.000
Demo1_Operator	0.000
Demo1_Result	0.000
Demo1_Input2_Python	45.000

Figure 46-The result for modifying code

6.3 Example for modifying Python code in ModelX Page 1

This is example for modify code in Python code. You can modify code for your purpose.

a) Open **ModelX_Page1.xml** file to edit, you can find this file at the location as below.

$(DocumentPath)\ EngineLab\EL Console\Plugins\EL_PythonQt\ModelX\ModelX_Page1.xml

Find lines of code in **ModelX_Page1.xml** file:

```
<control type="combobox" rect="30 125 230 25">
            <attr name="id" value="28" />
            <attr name="item" value="axis_value * 0.1" />
            <attr name="item" value="axis_value * 10" />
</control
```

b) Modify code as below.

```
<control type="combobox" rect="30 125 230 25">
            <attr name="id" value="28" />
            <attr name="item" value="axis_value * 0.2" />
            <attr name="item" value="axis_value * 20" />
</control
```

c) Open **Py_ModelX_Page1.py** file to edit, you can find this file at the location as below.

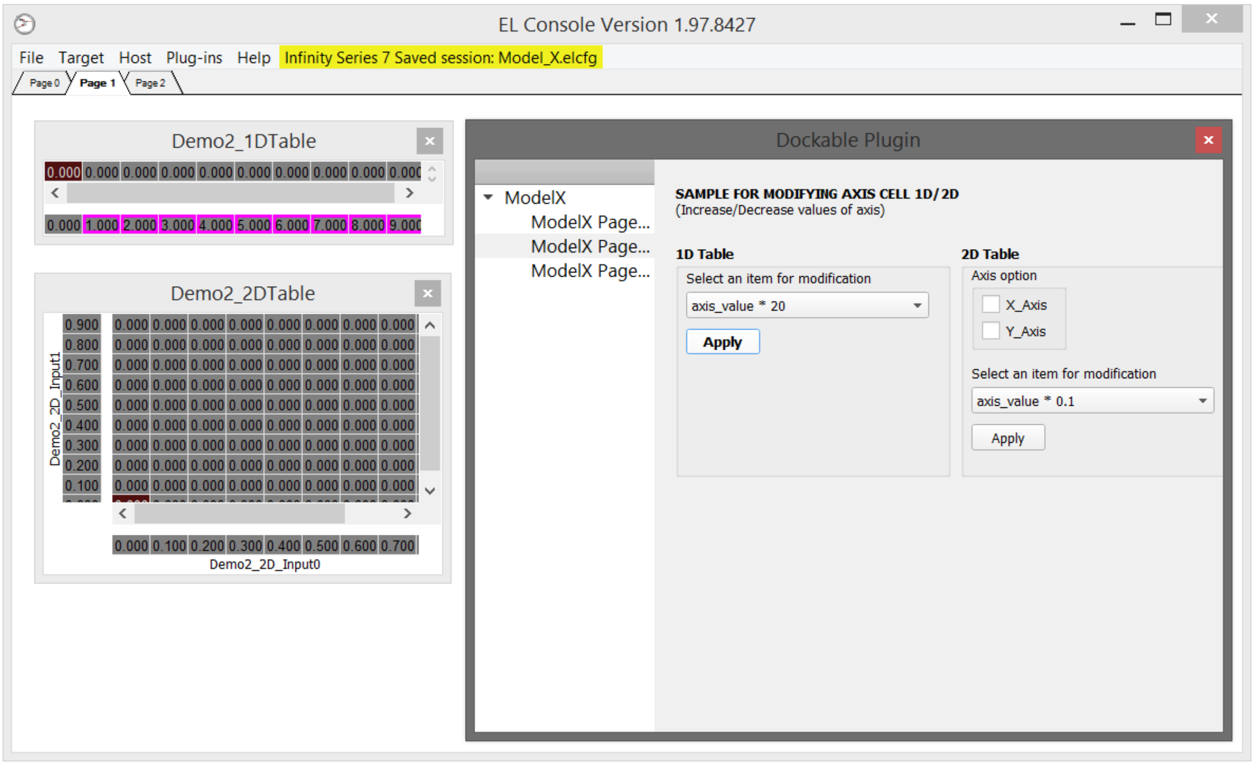
$(DocumentPath)\EngineLab\EL Console\Plugins\ EL_PythonQt\ModelX \scripts\Py_ModelX_Page1.py

Find lines of code in **Py_ModelX_Page1.py** file:

```
…
if comboboxValue.value == "axis_value * 0.2":
  for x in range(0, 10):
     xAxisValue = EL_TableChannelValue("Demo2_1DTable", INFOSELECT_X_AXIS, x)
     EL_WriteTableChannel("Demo2_1DTable", INFOSELECT_X_AXIS, x, xAxisValue * 0.2)
elif comboboxValue.value == "axis_value * 20":
  for x in range(0, 10):
     xAxisValue = EL_TableChannelValue("Demo2_1DTable", INFOSELECT_X_AXIS, x)
     EL_WriteTableChannel("Demo2_1DTable", INFOSELECT_X_AXIS, x, xAxisValue * 20)
```

d) Modify 0.1 -> 0.2 and 10 -> 20

e) Reload the plugin and check “Increase/Decrease values of axis”.


EL Console Version 1.97.8427
File
Target
Host
Plug-ins
Help
Infinity Series 7 Saved session: Model_X.elcfg
Page 0
Page 1
Page 2

Demo2_1DTable

0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.00
0.000	1.000	2.000	3.000	4.000	5.000	6.000	7.000	8.000	9.00

Demo2_2DTable

Demo2_2D_Input1	0.000	0.100	0.200	0.300	0.400	0.500	0.600	0.700
0.900	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.800	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.700	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.600	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.500	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.400	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.300	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.200	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.100	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Demo2_2D_Input0

Dockable Plugin

- ModelX
  - ModelX Page...
  - ModelX Page...
  - ModelX Page...

SAMPLE FOR MODIFYING AXIS CELL 1D/2D
(Increase/Decrease values of axis)

1D Table
Select an item for modification
axis_value * 20
Apply

2D Table
Axis option
☐ X_Axis
☐ Y_Axis
Select an item for modification
axis_value * 0.1
Apply

Figure 47-The result for modifying code

6.4 Example for modifying Python code in ModelX Page 2

This is example for modify code in Python code. You can modify code for your purpose.

a) Open **ModelX_Page2.xml** file to edit, you can find this file at the location as below

$(DocumentPath)\ EngineLab\EL Console\Plugins\EL_PythonQt\ModelX\ModelX_Page2.xml

Find lines of code in **ModelX_Page2.xml** file:

```
<control type="combobox" rect="30 170 220 25">
            <attr name="id" value="43" />
            <attr name="item" value="table_math_00" />
            <attr name="item" value="table_math_01" />
            <attr name="item" value="table_math_02" />
            <attr name="item" value="table_math_03" />
            <attr name="item" value="table_math_04" />
            <attr name="item" value="table_math_05" />
            <attr name="item" value="table_math_06" />
            <attr name="item" value="table_math_07" />
            <attr name="item" value="table_math_08" />
            <attr name="item" value="table_math_09" />
 </control>
```

b) Add an item (**table_math_10**) as code below

```
<control type="combobox" rect="30 170 220 25">
            <attr name="id" value="43" />
            <attr name="item" value="table_math_00" />
            <attr name="item" value="table_math_01" />
            <attr name="item" value="table_math_02" />
            <attr name="item" value="table_math_03" />
            <attr name="item" value="table_math_04" />
            <attr name="item" value="table_math_05" />
            <attr name="item" value="table_math_06" />
            <attr name="item" value="table_math_07" />
            <attr name="item" value="table_math_08" />
            <attr name="item" value="table_math_09" />
            <attr name="item" value="table_math_10" />
 </control>
```

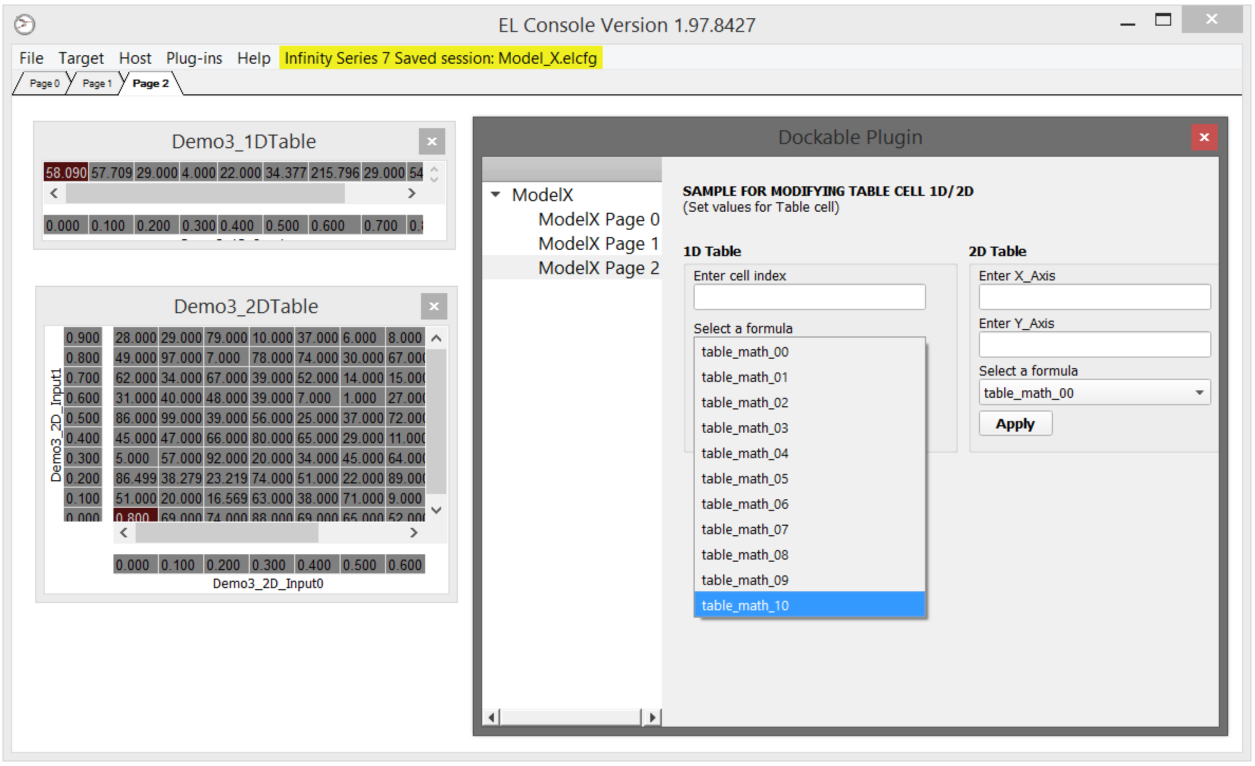
c) Open **Py_ModelX_Page2.py** file to edit, you can find this file at the location as below.

$(DocumentPath)\EngineLab\EL Console\Plugins\ EL_PythonQt\ModelX \scripts\Py_ModelX_Page2.py

Find lines of code in **Py_ModelX_Page2.py** file, and add code as below:

```
elif (comboboxValue.value == "table_math_07"):
        #Modify the current cell value(s) by 1
        primaryValue = primaryValue - 1
elif (comboboxValue.value == "table_math_08"):
        #Modify the current cell value(s) by 10
        primaryValue = primaryValue + 10
elif (comboboxValue.value == "table_math_09"):
        #Modify the current cell value(s) by 10
        primaryValue = primaryValue - 10
elif (comboboxValue.value == "table_math_10"):
        #Modify the current cell value(s) by 10
        primaryValue = primaryValue * 10
```

d) Save the files (.xml and .py). Start EL Console -> load session and layout file -> click on combobox control, the screen is displayed picture as below.



Demo3_1DTable

58.090	57.709	29.000	4.000	22.000	34.377	215.796	29.000	54
0.000	0.100	0.200	0.300	0.400	0.500	0.600	0.700	0.

Demo3_2DTable

Demo3_2D_Input1							
0.900	28.000	29.000	79.000	10.000	37.000	6.000	8.000
0.800	49.000	97.000	7.000	78.000	74.000	30.000	67.00
0.700	62.000	34.000	67.000	39.000	52.000	14.000	15.00
0.600	31.000	40.000	48.000	39.000	7.000	1.000	27.00
0.500	86.000	99.000	39.000	56.000	25.000	37.000	72.00
0.400	45.000	47.000	66.000	80.000	65.000	29.000	11.00
0.300	5.000	57.000	92.000	20.000	34.000	45.000	64.00
0.200	86.499	38.279	23.219	74.000	51.000	22.000	89.00
0.100	51.000	20.000	16.569	63.000	38.000	71.000	9.000
0.000	0.800	69.000	74.000	88.000	69.000	65.000	52.00
	0.000	0.100	0.200	0.300	0.400	0.500	0.600

Demo3_2D_Input0

SAMPLE FOR MODIFYING TABLE CELL 1D/2D
(Set values for Table cell)

1D Table

Enter cell index

Select a formula

table_math_00
table_math_01
table_math_02
table_math_03
table_math_04
table_math_05
table_math_06
table_math_07
table_math_08
table_math_09
table_math_10

2D Table

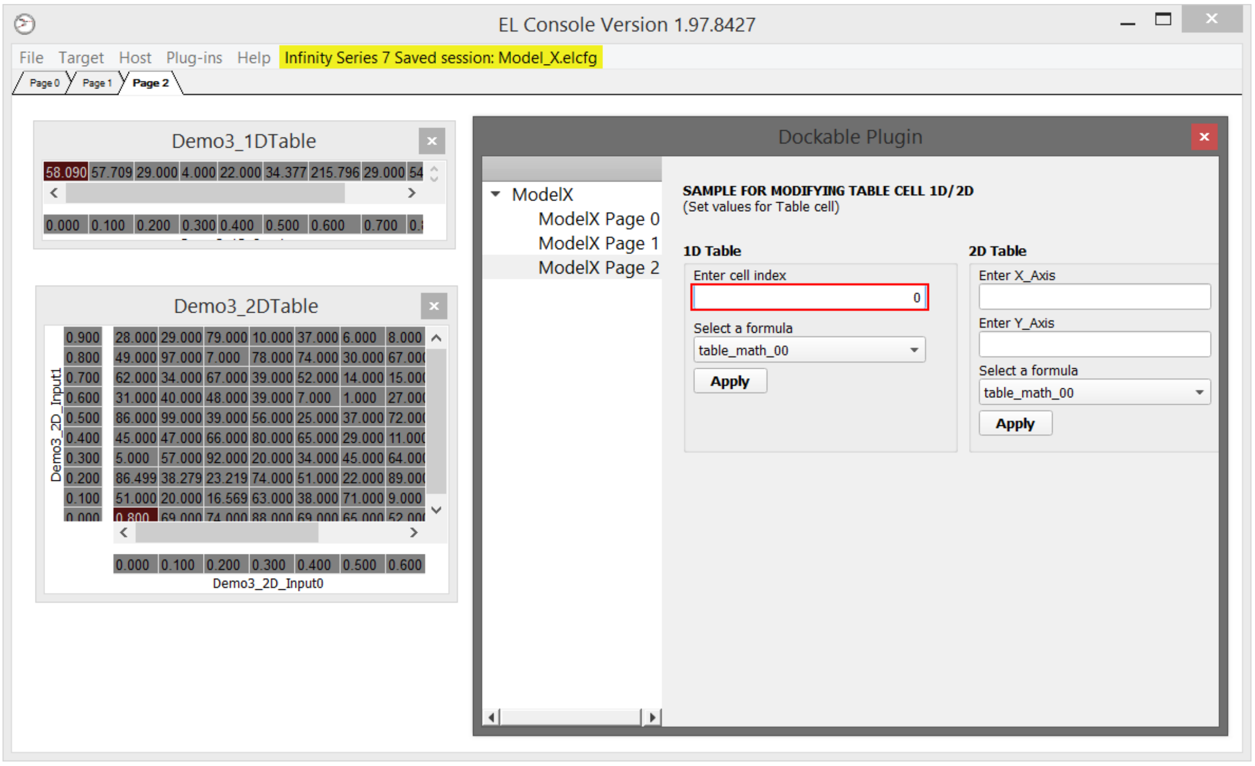
Enter X_Axis

Enter Y_Axis

Select a formula: table_math_00

Figure 48-Loading EL Console and Setting for the controls

e) Click on Apply button, check update value for table cell.



EL Console Version 1.97.8427

File Target Host Plug-ins Help Infinity Series 7 Saved session: Model_X.elcfg

Page 0 Page 1 Page 2

Demo3_1DTable

58.090	57.709	29.000	4.000	22.000	34.377	215.796	29.000	54
0.000	0.100	0.200	0.300	0.400	0.500	0.600	0.700	0.

Demo3_2DTable

Demo3_2D_Input1							
0.900	28.000	29.000	79.000	10.000	37.000	6.000	8.000
0.800	49.000	97.000	7.000	78.000	74.000	30.000	67.00
0.700	62.000	34.000	67.000	39.000	52.000	14.000	15.00
0.600	31.000	40.000	48.000	39.000	7.000	1.000	27.00
0.500	86.000	99.000	39.000	56.000	25.000	37.000	72.00
0.400	45.000	47.000	66.000	80.000	65.000	29.000	11.00
0.300	5.000	57.000	92.000	20.000	34.000	45.000	64.00
0.200	86.499	38.279	23.219	74.000	51.000	22.000	89.00
0.100	51.000	20.000	16.569	63.000	38.000	71.000	9.000
0.000	0.800	69.000	74.000	88.000	69.000	65.000	52.00
	0.000	0.100	0.200	0.300	0.400	0.500	0.600

Demo3_2D_Input0

Dockable Plugin

ModelX
ModelX Page 0
ModelX Page 1
ModelX Page 2

SAMPLE FOR MODIFYING TABLE CELL 1D/2D
(Set values for Table cell)

1D Table

Enter cell index: 0

Select a formula: table_math_00

Apply

2D Table

Enter X_Axis:

Enter Y_Axis:

Select a formula: table_math_00

Apply

Figure 49-The result for modifying code

6.5 Example for modifying Python test Qt control functions

Suppose you want to access and configure Qt control. You have to modify Python code and XML file; you follow the steps as below.

a) Open **PyScriptTestQtControl.py** file to edit, you can find this file at the location as below.

$(DocumentPath)\EngineLab\EL Console\Plugins\ EL_PythonQt\ModelX \scripts\PyScriptTestQtControl.py

Find lines of code in **PyScriptTestQtControl.py** file:

```
g_idEditbox = "20"

def EditboxControlTest():
```

b) Change text "**20**" to "**6**" as image below. This number is an attribute id you have defined for a Qt control.

```
g_idEditbox = "6"

def EditboxControlTest():
```

c) Open **ModelX_Page0.xml** file, you can find this file at the location as below.

$(DocumentPath)\ EngineLab\EL Console\Plugins\EL_PythonQt\ModelX\ModelX_Page0.xml

Find the lines of code in **ModelX_Page0.xml** file:

```
<control type="button" rect="160 325 80 24">
            <attr name="id" value="17" />
            <attr name="caption" value="Apply" />
            <attr name="style" value="bold" />
            <attr name="clicked_event" value="demo1 apply" />
 </control>
```

d) Change text "**demo1 apply**" to "**PyScriptTestQtControl EditboxControlTest**" as image below

```
<control type="button" rect="160 325 80 24">
            <attr name="id" value="17" />
            <attr name="caption" value="Apply" />
            <attr name="style" value="bold" />
            <attr name="clicked_event" value="PyScriptTestQtControl EditboxControlTest " />
 </control>
```

e) Save the files (.xml and .py). Start EL Console, load session and plugin, the screen is displayed picture as below.

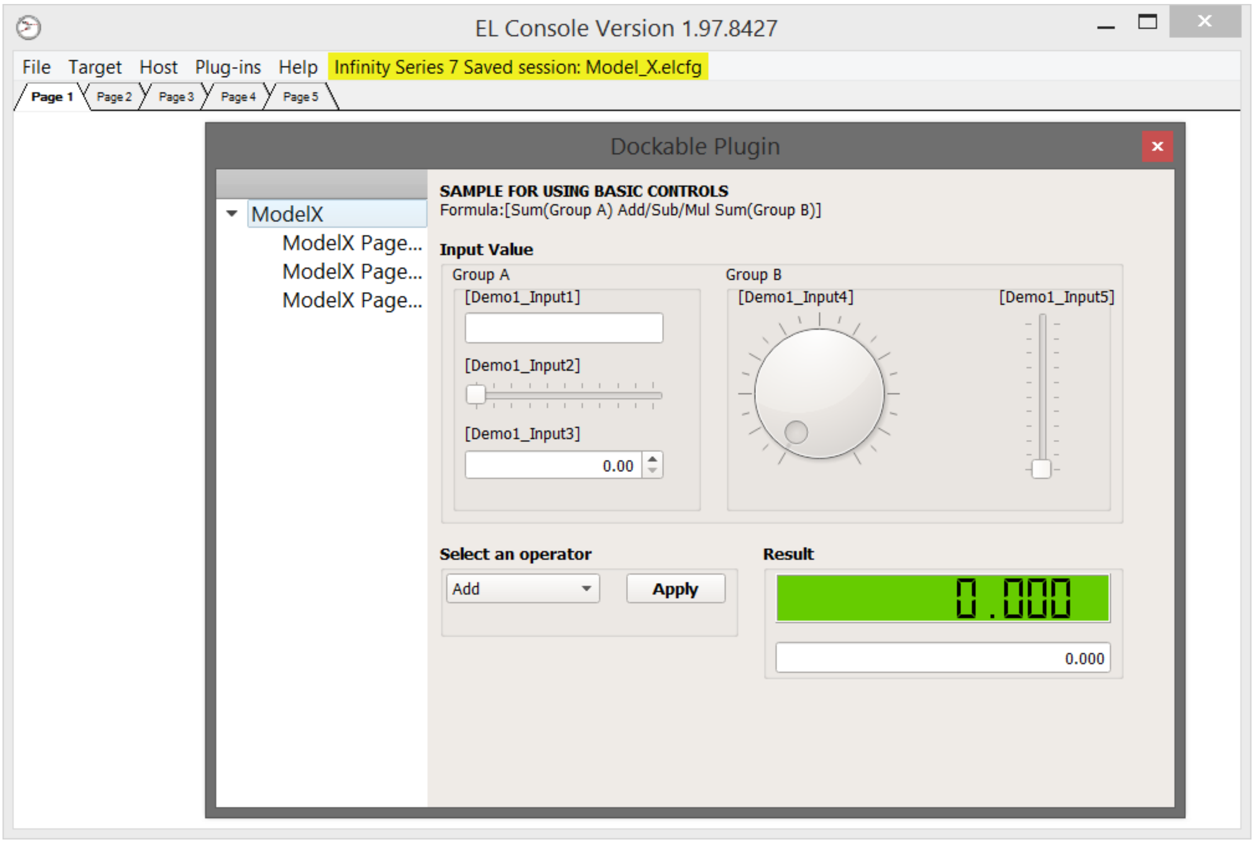


Figure 50-The result for modifying code

f) Click on Apply button to execute all control function.

7. How to use debugger window

The plugin provides a debugging dialog which uses to re-direct error output and use to print string for debugging.

7.1 Re-direct error output

Whenever you modify and program python code, the plugin provide a way to inform if python code has error.

a) Open **Py_ModelX_Page0.py** file to edit, you can find this file at the location as below.

$(DocumentPath)\EngineLab\EL Console\Plugins\ EL_PythonQt\ModelX \scripts\Py_ModelX_Page0.py

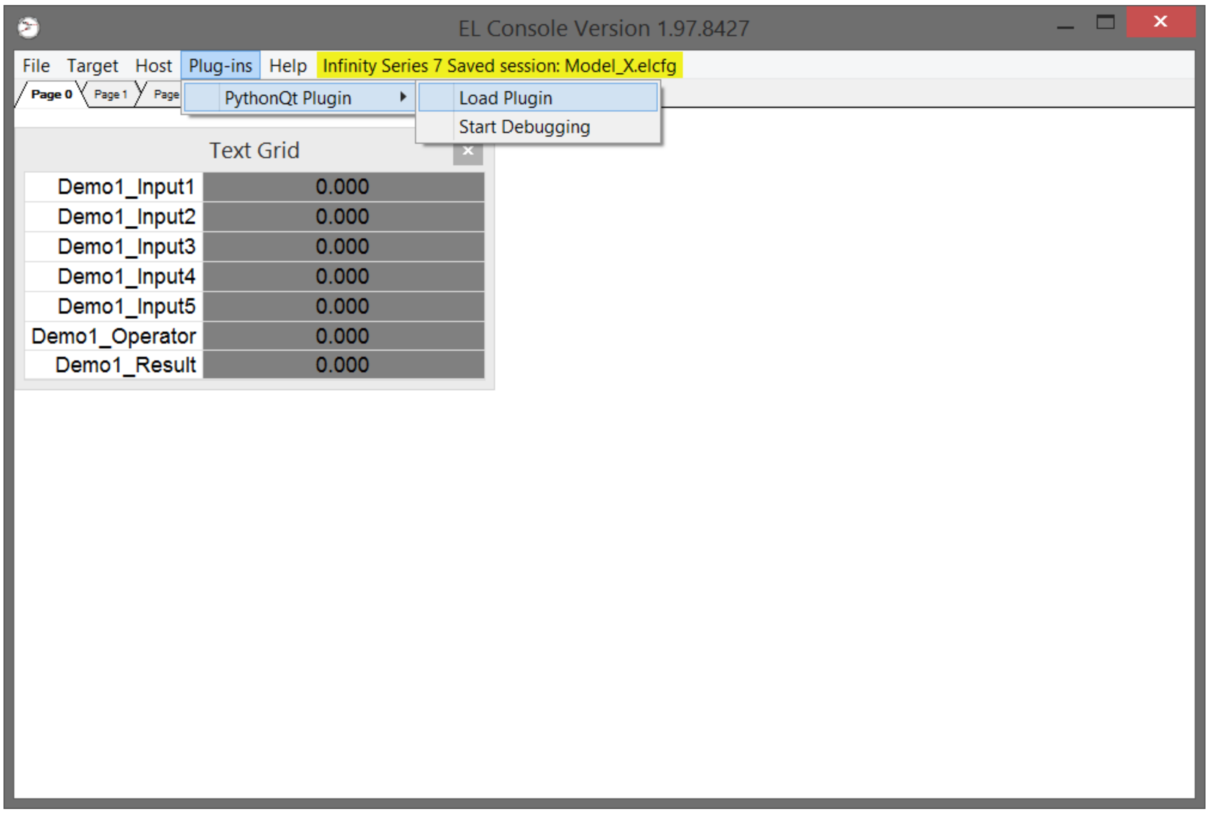
Find lines of code in **Py_ModelX_Page0.py** file:

```
objEdit = EditControl()

objSlider = SliderControl()

….
```

b) Comment text "**objSlider = SliderControl()**" as image below

```
objEdit = EditControl()

#objSlider = SliderControl()
```

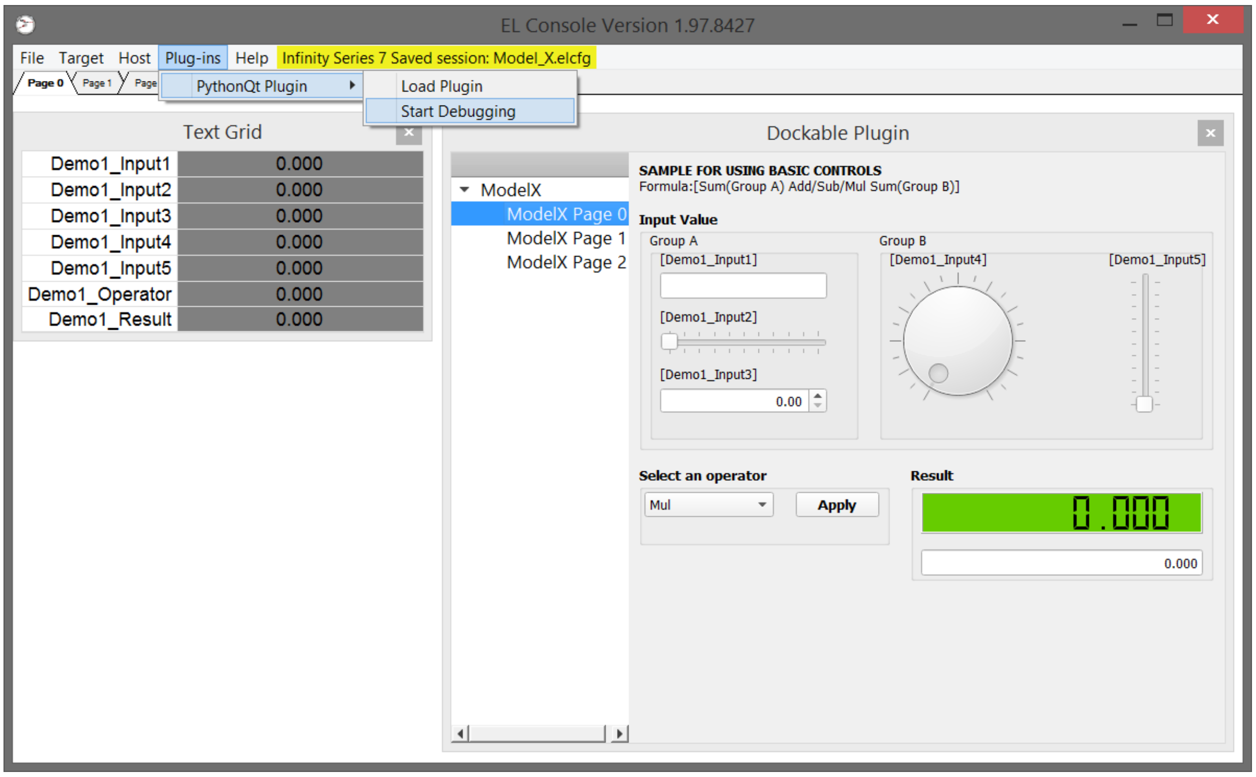
i) Save the files (.xml and .py). Start EL Console. Select Plug-ins-> PythonQt Plugin->Load Plugin.



Demo1_Input1	0.000
Demo1_Input2	0.000
Demo1_Input3	0.000
Demo1_Input4	0.000
Demo1_Input5	0.000
Demo1_Operator	0.000
Demo1_Result	0.000

Figure 51-Load the Plugin

d) Select Plug-ins-> PythonQt Plugin->Start Debugging.

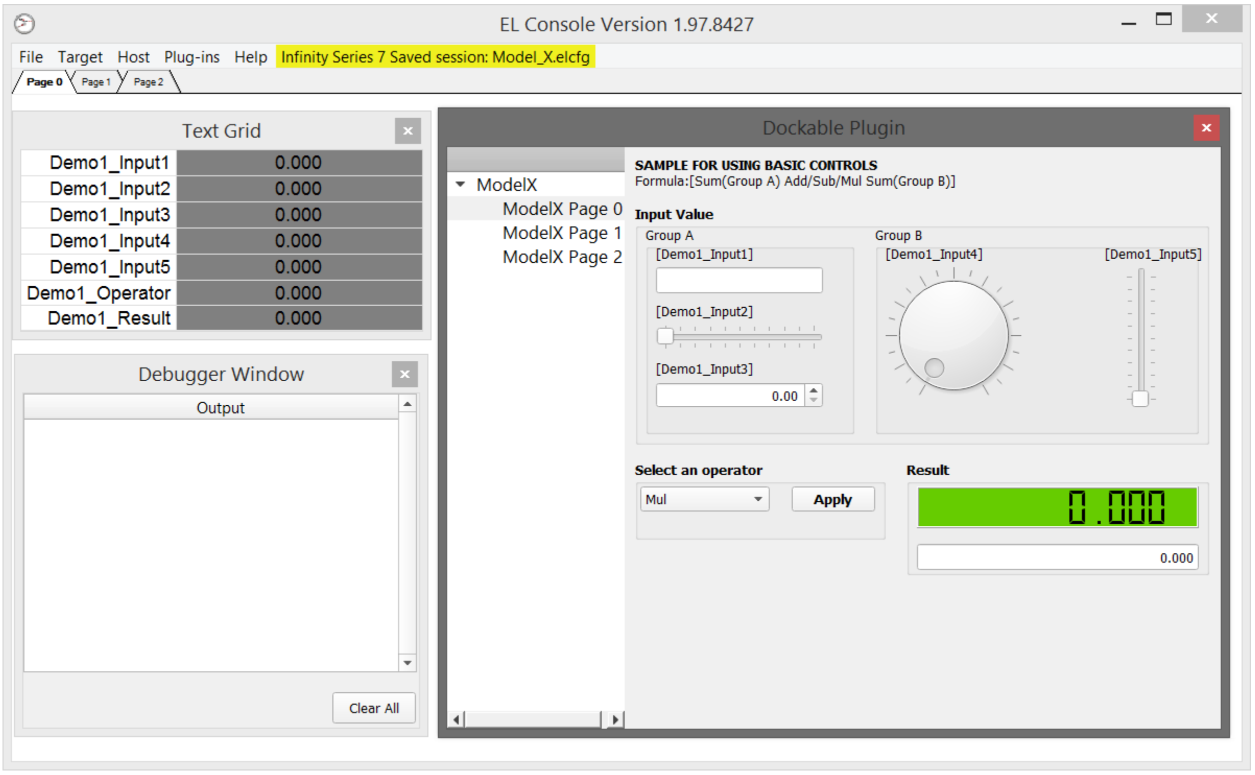


Demo1_Input1	0.000
Demo1_Input2	0.000
Demo1_Input3	0.000
Demo1_Input4	0.000
Demo1_Input5	0.000
Demo1_Operator	0.000
Demo1_Result	0.000

SAMPLE FOR USING BASIC CONTROLS
Formula:[Sum(Group A) Add/Sub/Mul Sum(Group B)]
Input Value
Group A: [Demo1_Input1], [Demo1_Input2], [Demo1_Input3] 0.00
Group B: [Demo1_Input4], [Demo1_Input5]
Select an operator: Mul
Result: 0.000 / 0.000

Figure 52- Start the Debugging

e) After load the plugin and the debugging.

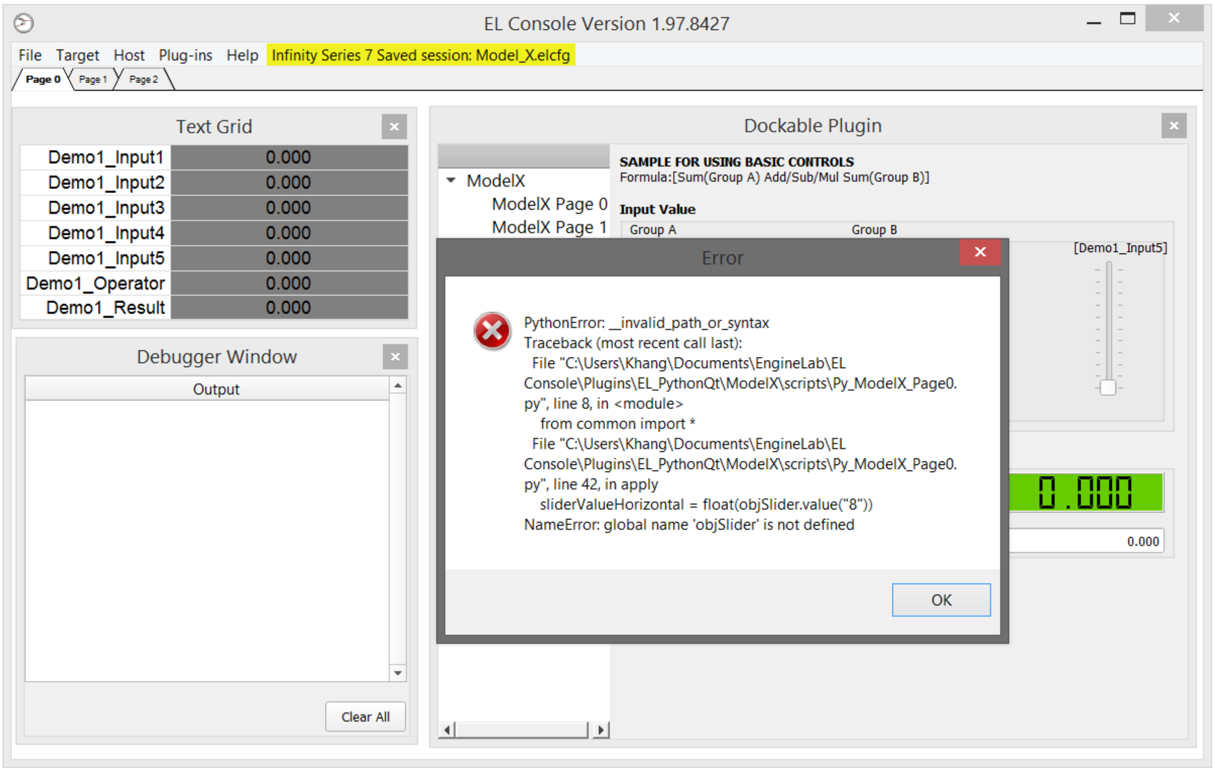


Demo1_Input1	0.000
Demo1_Input2	0.000
Demo1_Input3	0.000
Demo1_Input4	0.000
Demo1_Input5	0.000
Demo1_Operator	0.000
Demo1_Result	0.000

SAMPLE FOR USING BASIC CONTROLS
Formula:[Sum(Group A) Add/Sub/Mul Sum(Group B)]
Input Value
Group A: [Demo1_Input1], [Demo1_Input2], [Demo1_Input3] 0.00
Group B: [Demo1_Input4], [Demo1_Input5]
Select an operator: Mul
Result: 0.000 / 0.000

Figure 53- Start the Debugging

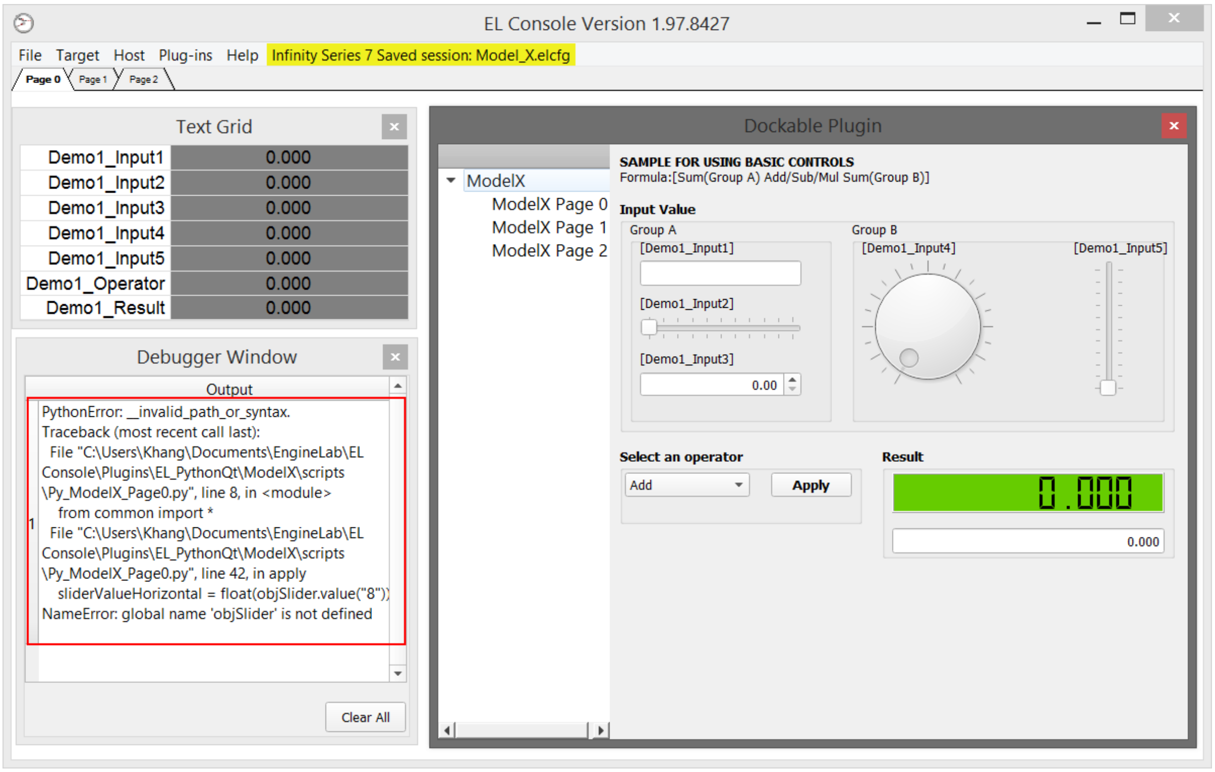
f) Click on apply button. A message box will inform an error and also the debugger window will print out the same error.


EL Console Version 1.97.8427
File Target Host Plug-ins Help Infinity Series 7 Saved session: Model_X.elcfg
Page 0 Page 1 Page 2
Text Grid
Debugger Window
Output
Clear All
Dockable Plugin
ModelX
ModelX Page 0
ModelX Page 1
SAMPLE FOR USING BASIC CONTROLS
Formula:[Sum(Group A) Add/Sub/Mul Sum(Group B)]
Input Value
Group A
Group B
[Demo1_Input5]
Error
OK

Demo1_Input1	0.000
Demo1_Input2	0.000
Demo1_Input3	0.000
Demo1_Input4	0.000
Demo1_Input5	0.000
Demo1_Operator	0.000
Demo1_Result	0.000

```
PythonError: _invalid_path_or_syntax
Traceback (most recent call last):
  File "C:\Users\Khang\Documents\EngineLab\EL
Console\Plugins\EL_PythonQt\ModelX\scripts\Py_ModelX_Page0.
py", line 8, in <module>
    from common import *
  File "C:\Users\Khang\Documents\EngineLab\EL
Console\Plugins\EL_PythonQt\ModelX\scripts\Py_ModelX_Page0.
py", line 42, in apply
    sliderValueHorizontal = float(objSlider.value("8"))
NameError: global name 'objSlider' is not defined
```

Figure 54- Message box error



Demo1_Input1	0.000
Demo1_Input2	0.000
Demo1_Input3	0.000
Demo1_Input4	0.000
Demo1_Input5	0.000
Demo1_Operator	0.000
Demo1_Result	0.000

```
PythonError: _invalid_path_or_syntax.
Traceback (most recent call last):
  File "C:\Users\Khang\Documents\EngineLab\EL
Console\Plugins\EL_PythonQt\ModelX\scripts
\Py_ModelX_Page0.py", line 8, in <module>
    from common import *
  File "C:\Users\Khang\Documents\EngineLab\EL
Console\Plugins\EL_PythonQt\ModelX\scripts
\Py_ModelX_Page0.py", line 42, in apply
    sliderValueHorizontal = float(objSlider.value("8"))
NameError: global name 'objSlider' is not defined
```

Figure 55- Error string in debugger window

7.2 Print string to debugger window

In python code the plugin provide a function “**printfString(text):**” to print string to debugger window. You can refer the operation of this function in 6.1.1 The functions for the Python script. To use this function you can refer to file “**Py_ModelX_Page0.py**”. You can find this file at the location as below.

$(DocumentPath)\EngineLab\EL Console\Plugins\ EL_PythonQt\ModelX \scripts\Py_ModelX_Page0.py

Find lines of code in **Py_ModelX_Page0.py** file:

```
….
EL_WriteNoOpChannel("Demo1_Result", totalValue)
printfString("After click on Apply button, the result is " + str(totalValue))
….
```

Also refer in 4.2.1 Testing for ModelX Page 0 to see how it works.

8. How to build source code of the Plugin

When you want to change code and rebuild the .dll file for the Plugin, you refer the "*How to build the Plugin.doc*" file.

9. How to create the installer for the Plugin

When you want to create new an installer for the Plugin after you update .xml/Python code/plugin dll, you refer the "*How to create installer.doc*" file.

REVISION HISTORY	DockablePluginPythonQt_Setup_Manual

Rev.	**Date**	**Description**	
		Page	**Summary**
1.00	December 15, 2015		Create new document
1.01	March 25, 2016		Update document
1.02	May 05, 2016		Update document

Plugin DLL for EL Console

ENGINELAB