Multiple Popups

Available Methods

There are many ways to open a new Display or Popup in a project. For each method, some specific characteristics must be considered. The list below shows some of those methods as well as a brief description of their behavior.

Display.Open

- Begins the Open Display process.
- The syntax is as follows:

Display.<DisplayName>.Open()

- Returns a flag (true or false) indicating success or fail.

E.g.: @Display.MainPage.Open()

Display.OpenModal

- Open Display as Modal.
- The syntax is as follows:

Display.<DisplayName>.OpenModal()

- Not available for Mono and HTML5
- Returns a flag (true or false) indicating success or fail.

E.g.: @Display.SelectPage.OpenModal()

Display.NewPopup

- Open a new one. If the Display is already open, a new instance will open.
- The syntax is as follows:

Display.<DisplayName>.NewPopup(input parameters)

- The accepted input item parameters are:

LabelList[label1=tag1; label2=tag2], Left, Top, Width, Height.

- Always returns an empty string, independent if it succeeds or E.g.:

@Display.MainPage.NewPopup(["VALUE = Integer1; NAME = ’Example’"])

Client.OpenDisplay

- Open a Display.
- The syntax is as follows:

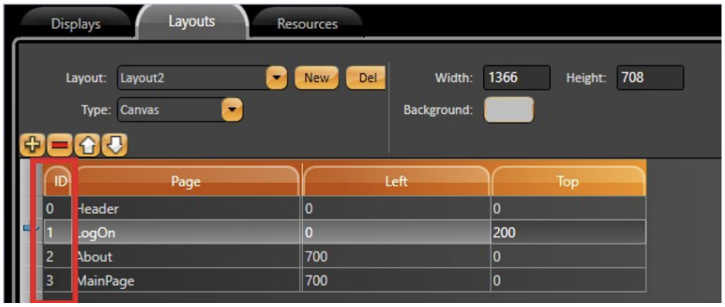
Client.OpenDisplay("<DisplayName>")

- Returns a flag (true or false) indicating success or fail.

E.g.: @Client.OpenDisplay("About")

Client.OpenDisplayAtIndex

- Open Display at x-order. List of Current Display is closed.



Layout: Layout2

Type: Canvas

Width: 1366

Height: 708

Background:

ID	Page	Left	Top
0	Header	0	0
1	LogOn	0	200
2	About	700	0
3	MainPage	700	0

- The syntax is as follows:

Client.OpenDisplayAtIndex("<DisplayName>", index)

- Returns a flag (true or false) indicating success or fail.

E.g.: @Client.OpenDisplayAtIndex("LogOn", 2)

Client.NewPopup

- Open a new one. If the Display is already open, a new instance will open.
- The syntax is as follows:

Client.NewPopup(input parameters)

- The accepted input item parameters are:

LabelList[label1=tag1; label2=tag2], Left, Top, Width, Height.

- Always returns an empty string, independent if it succeeds or fail.

E.g.: @Client.NewPopup("GAUGE = Integer2")

Differences

When you add an OpenDisplay method to a project but then change the name of the Display, you will have some different results if you use the **@Client** or **@Display** methods.

- In the **@Display** methods: Since the DisplayName is used as one of the properties, the software is smart enough to detect if the Display Name was changed and automatically update the method.
- In the **@Client** methods: The DisplayName is used a string type input parameter. If the display name is changed, the user must manually update the method.

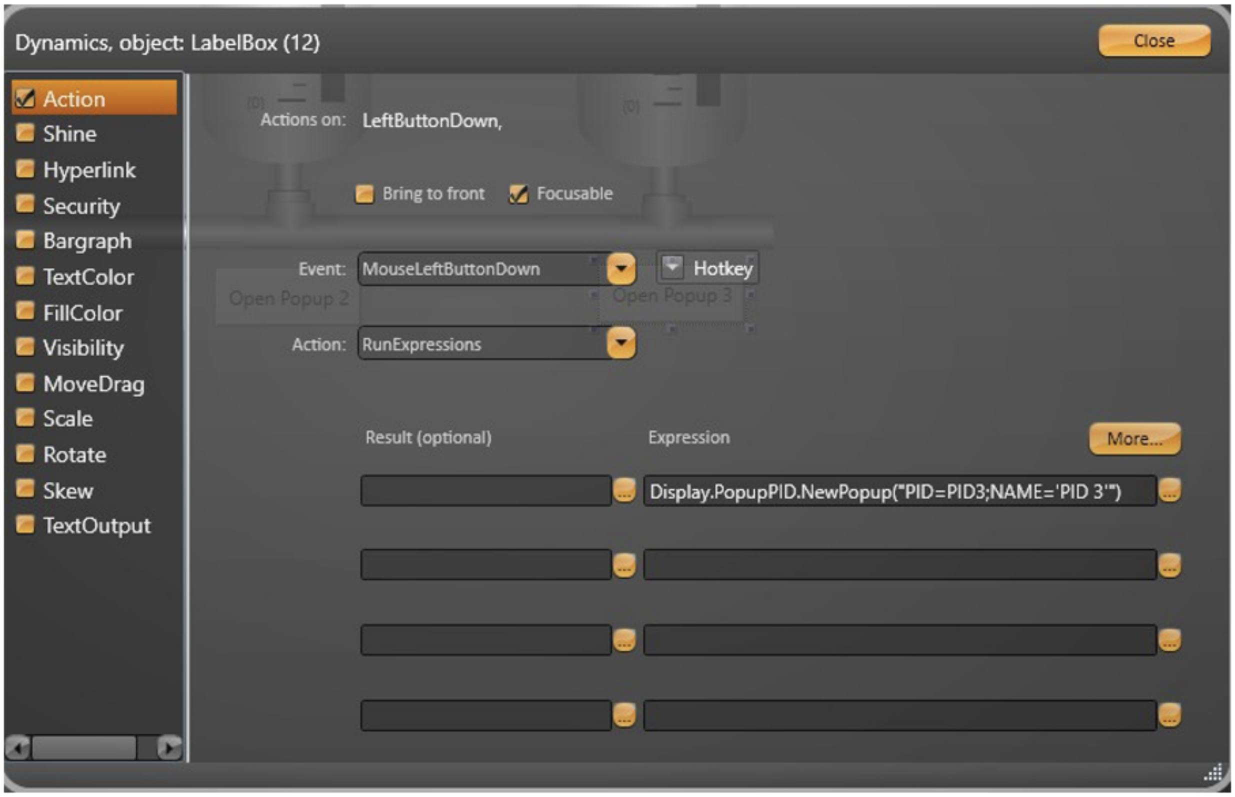
Demo Project and *@Client.NewPopup* Method

On the MainPage, you can find three buttons. When Popup1 or Popup2 is pressed, it runs a script in the CodeBehind as shown in the image below.

```
public void NewPopup1(object sender, System.Windows.Input.InputEventArgs e)
{
@Client.NewPopup("PopupPID","PID  =  PID1;  NAME  =  ’PID  1’");
}

public void NewPopup2(object sender, System.Windows.Input.InputEventArgs e)
{
@Client.NewPopup("PopupPID","PID  =  PID2  ;  NAME  =  ’PID  2’");
}
```

The button for Popup3 runs the expression shown in the image below:



Dynamics, object: LabelBox (12)

Actions on: LeftButtonDown,

☐ Bring to front ☑ Focusable

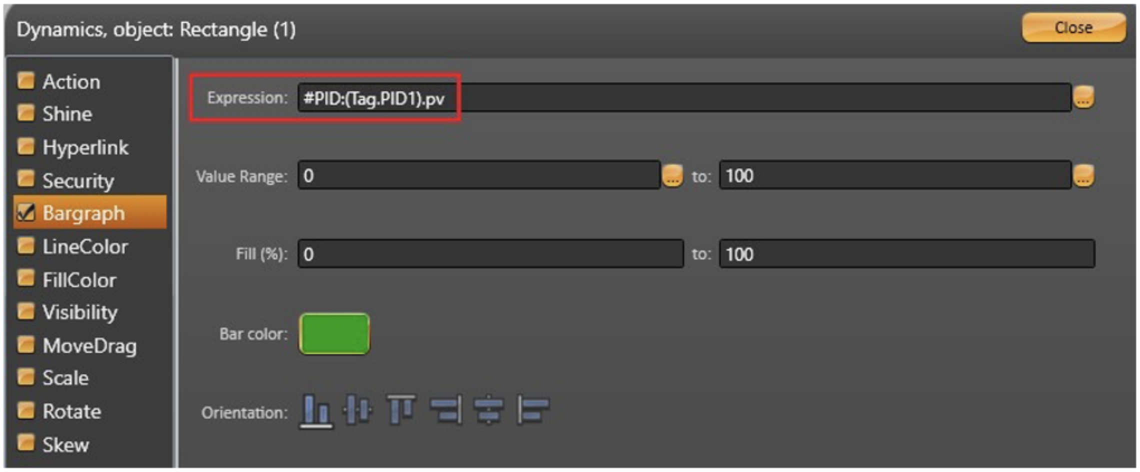
Event: MouseLeftButtonDown

Action: RunExpressions

Result (optional)	Expression
	Display.PopupPID.NewPopup("PID=PID3;NAME='PID 3'")

We can see the object items used as input parameters. The '***PopupPID***' is the Display Name. The '***PID***' is the label that will be used in the display elements, and the label '***Name***' is the text that will appear in the Popup Title Bar.

Looking into the PopupPID Display: If we open the Dynamics window for the Rectangles, we can see how the input parameter is passed to the object.



Dynamics, object: Rectangle (1)

Expression: #PID:(Tag.PID1).pv

Value Range: 0 to: 100

Fill (%): 0 to: 100

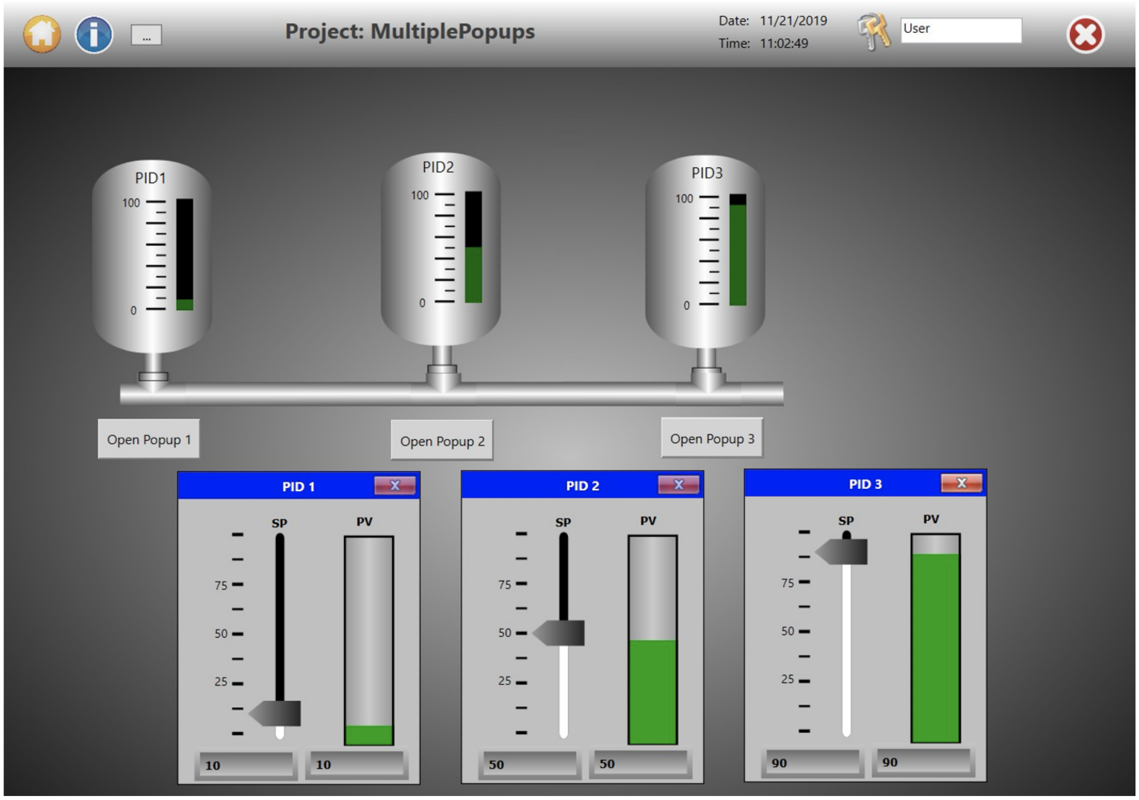
Bar color: 🟩

The '***#Label:***' is used to replace whatever is inside the parentheses with the inputted label.

The first Label, '***PID = PID Tag***', inserts the Tag into the element (as the image above), and '***Name = PID Name***' is configured in the CodeBehind, as the image below.

```
public void DisplayOpening()
{
    CurrentDisplay.SetTitle(  CurrentDisplay.GetLabel("NAME")  );
}
```

Finally, the project behavior in runtime is presented below.



NewPopup Characteristics

Symbols

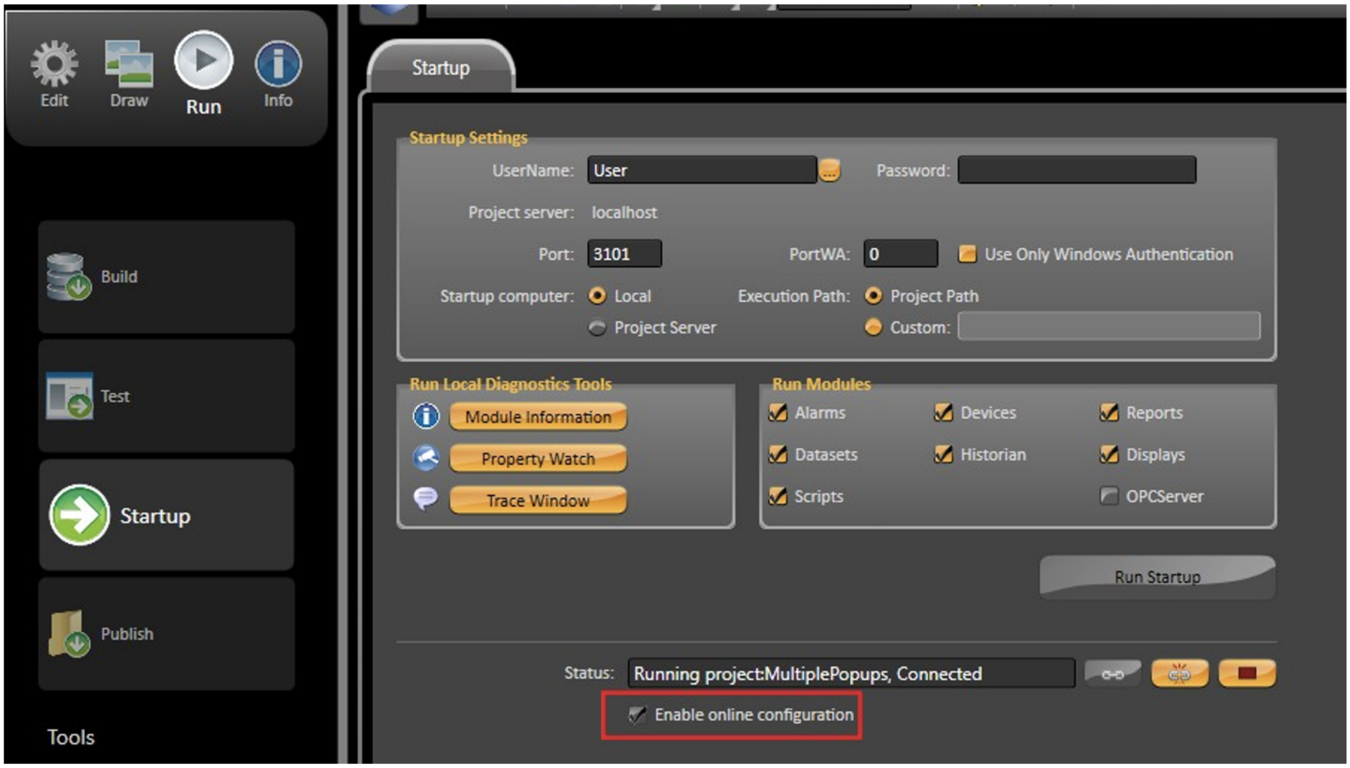
Labels are often used as parameters for symbols as a way to facilitate the project setup. However, there is a limitation when combining the NewPopup input labels and Symbols added to the Display.

If there are two or more of the same symbol in the display, the software will not differentiate them. Since all the labels are equal, it will associate all symbols to the same Tag.

Therefore, the utilization of multiple Popups, one for each symbol, is an excellent approach to achieve this goal.

Online Configuration

Online Configuration is another interesting feature to use in a project. It allows the modifications that are made to a project in the Engineering Environment to affect the application in real time. You can find it in ***Run > Test or Run > Startup***.



Startup Settings

UserName: User
Password:
Project server: localhost
Port: 3101
PortWA: 0
□ Use Only Windows Authentication
Startup computer: ● Local ○ Project Server
Execution Path: ● Project Path ○ Custom:

Run Modules

☑ Alarms ☑ Devices ☑ Reports
☑ Datasets ☑ Historian ☑ Displays
☑ Scripts □ OPCServer

Status: Running project:MultiplePopups, Connected

☑ Enable online configuration