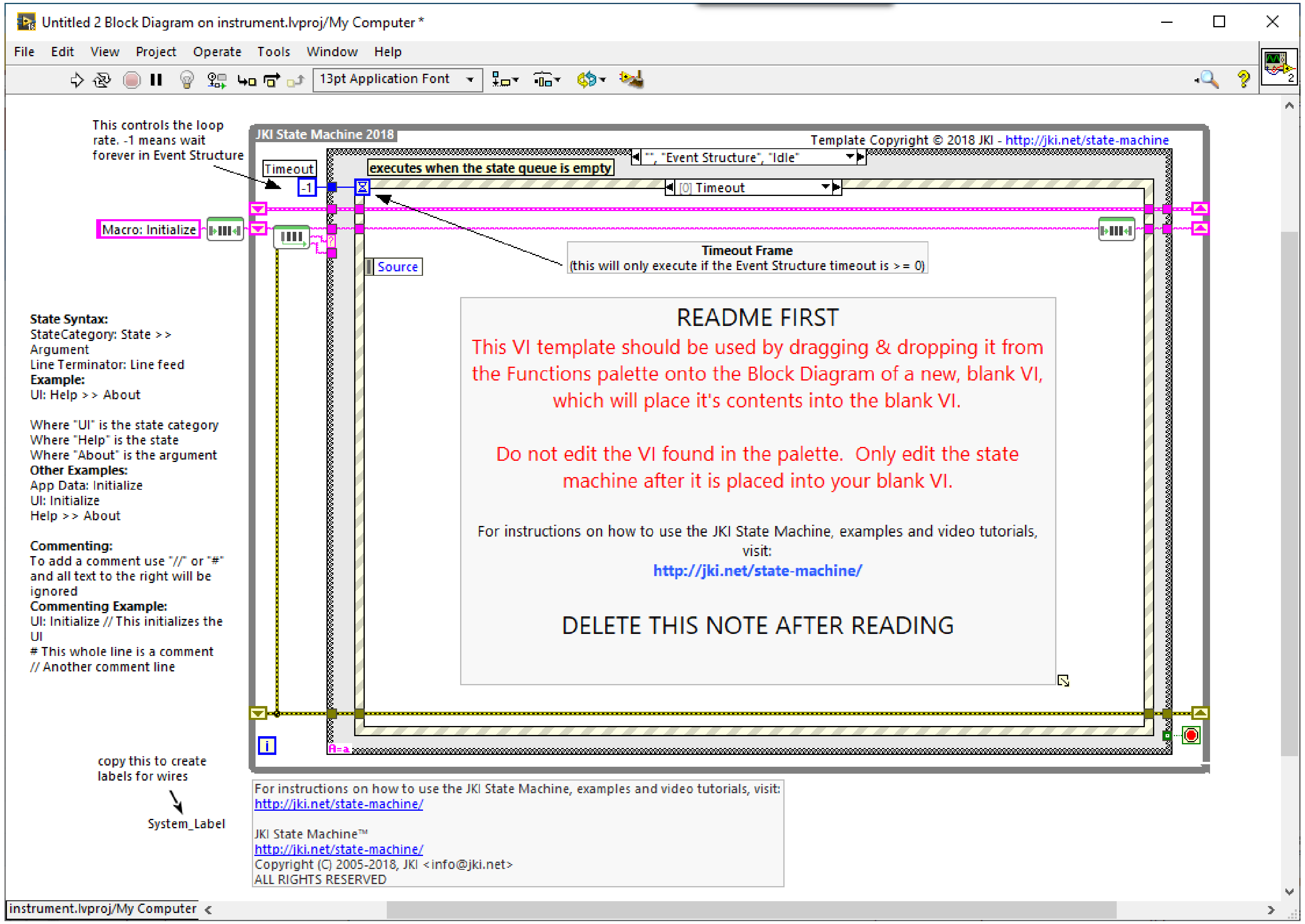
JKI State Machine Objects: Quick Introduction

Patrick Irvin, PhD
Department of Physics and Astronomy
University of Pittsburgh

JKI State Machines

JKI State Machine



Untitled 2 Block Diagram on instrument.lvproj/My Computer *

File Edit View Project Operate Tools Window Help

13pt Application Font

This controls the loop rate. -1 means wait forever in Event Structure

Macro: Initialize

State Syntax:
StateCategory: State >> Argument
Line Terminator: Line feed
Example:
UI: Help >> About

Where "UI" is the state category
Where "Help" is the state
Where "About" is the argument
Other Examples:
App Data: Initialize
UI: Initialize
Help >> About

Commenting:
To add a comment use "//" or "#" and all text to the right will be ignored
Commenting Example:
UI: Initialize // This initializes the UI
This whole line is a comment
// Another comment line

copy this to create labels for wires

System_Label

JKI State Machine 2018

Template Copyright © 2018 JKI - http://jki.net/state-machine

Timeout

executes when the state queue is empty

"", "Event Structure", "Idle"

[0] Timeout

Source

Timeout Frame
(this will only execute if the Event Structure timeout is >= 0)

README FIRST

This VI template should be used by dragging & dropping it from the Functions palette onto the Block Diagram of a new, blank VI, which will place it's contents into the blank VI.

Do not edit the VI found in the palette. Only edit the state machine after it is placed into your blank VI.

For instructions on how to use the JKI State Machine, examples and video tutorials, visit:
http://jki.net/state-machine/

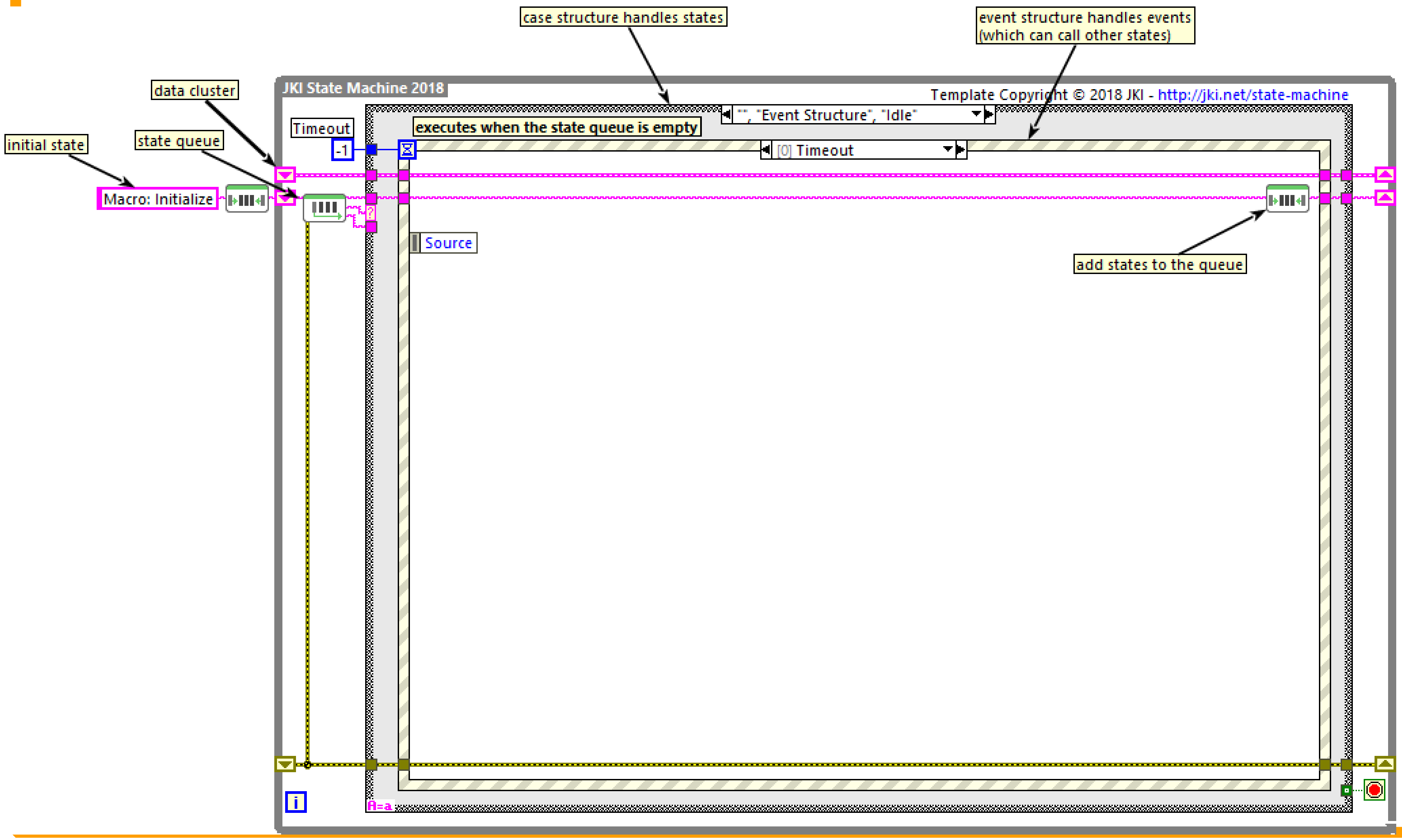
DELETE THIS NOTE AFTER READING

For instructions on how to use the JKI State Machine, examples and video tutorials, visit:
http://jki.net/state-machine/

JKI State Machine™
http://jki.net/state-machine/
Copyright (C) 2005-2018, JKI <info@jki.net>
ALL RIGHTS RESERVED

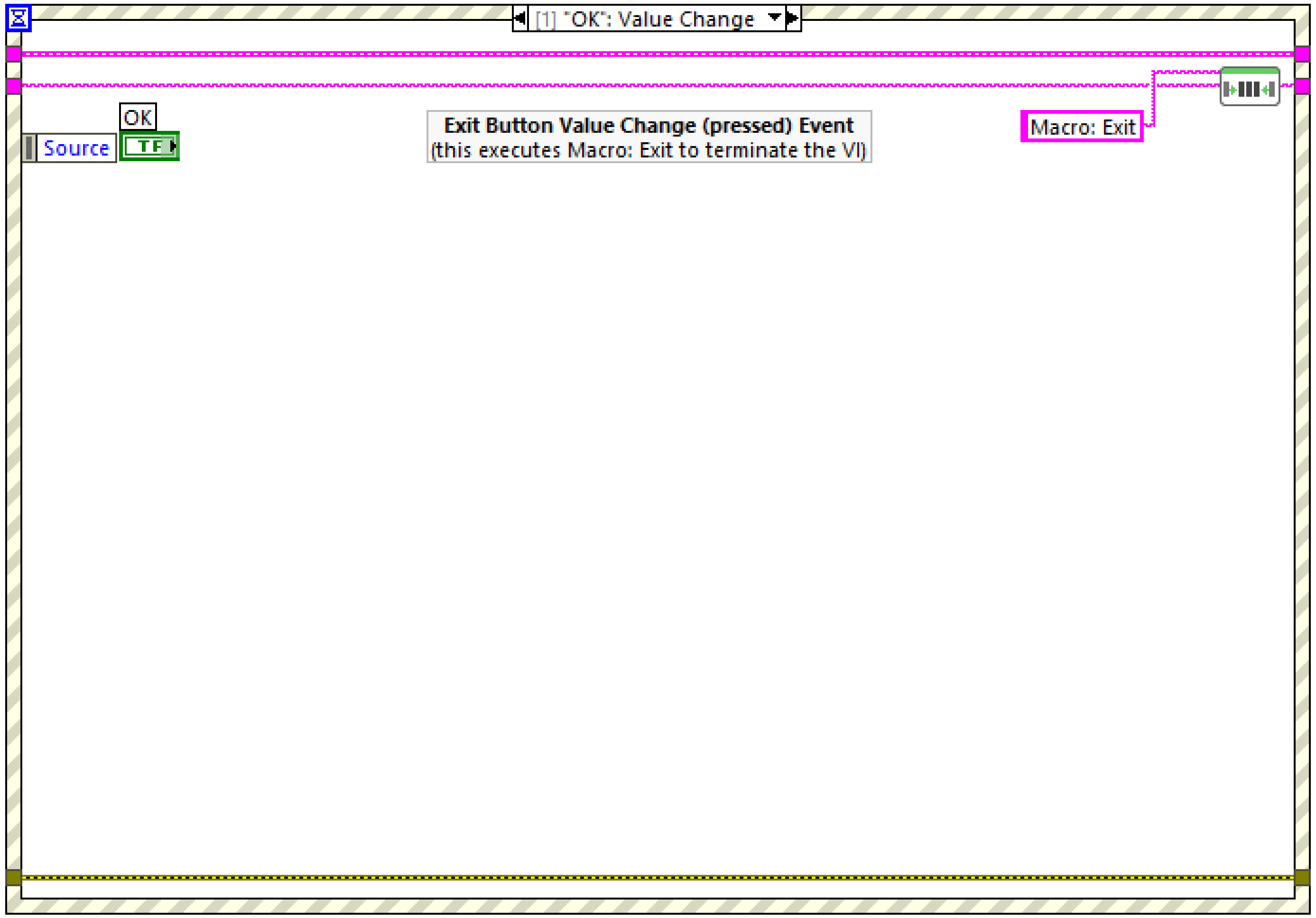
instrument.lvproj/My Computer

JKI State Machine



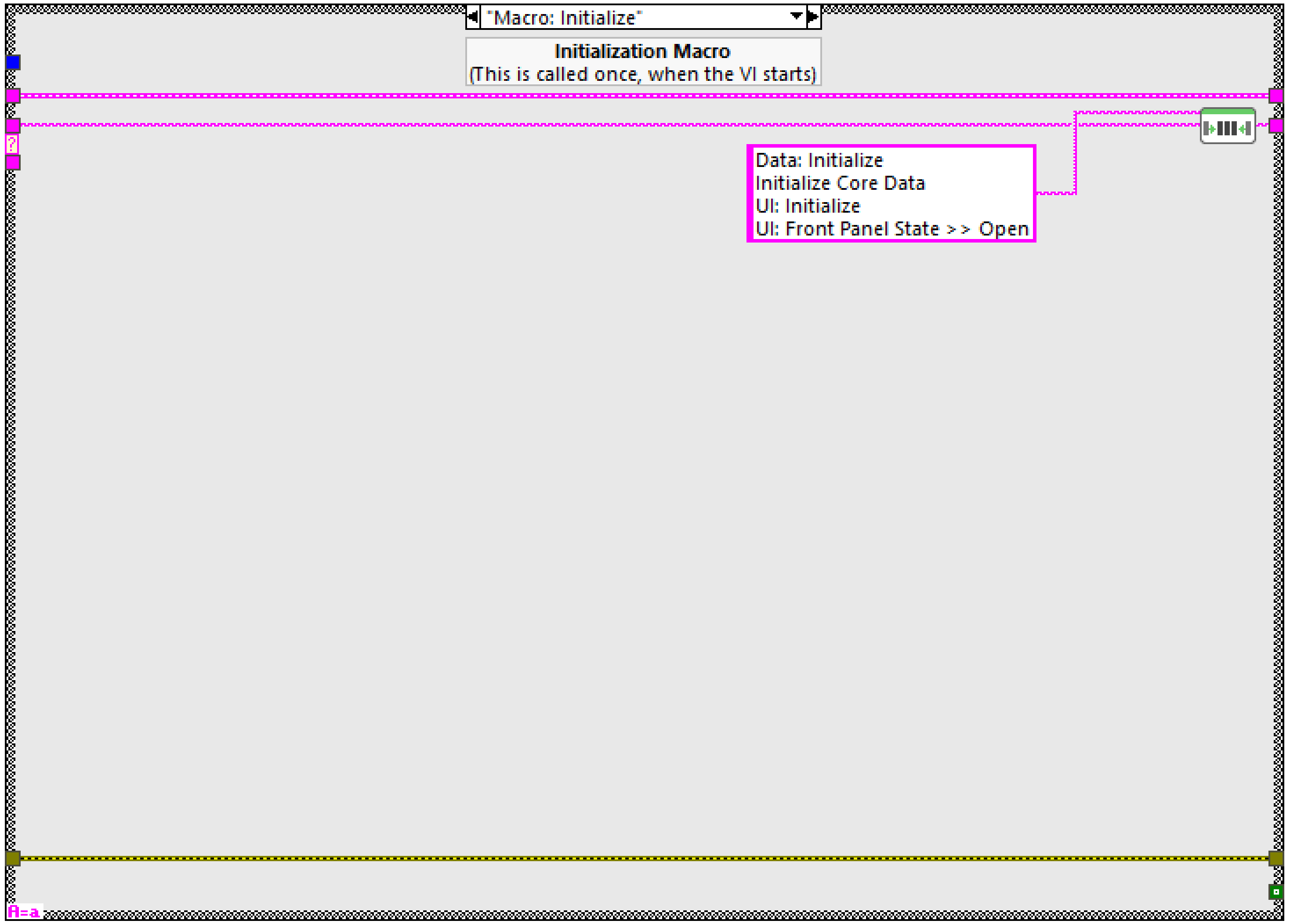
case structure handles states
event structure handles events (which can call other states)
JKI State Machine 2018
Template Copyright © 2018 JKI - http://jki.net/state-machine
data cluster
initial state
state queue
Timeout
-1
executes when the state queue is empty
"", "Event Structure", "Idle"
[0] Timeout
Macro: Initialize
Source
add states to the queue

Event example:



Exit Button Value Change (pressed) Event
(this executes Macro: Exit to terminate the VI)

State example:

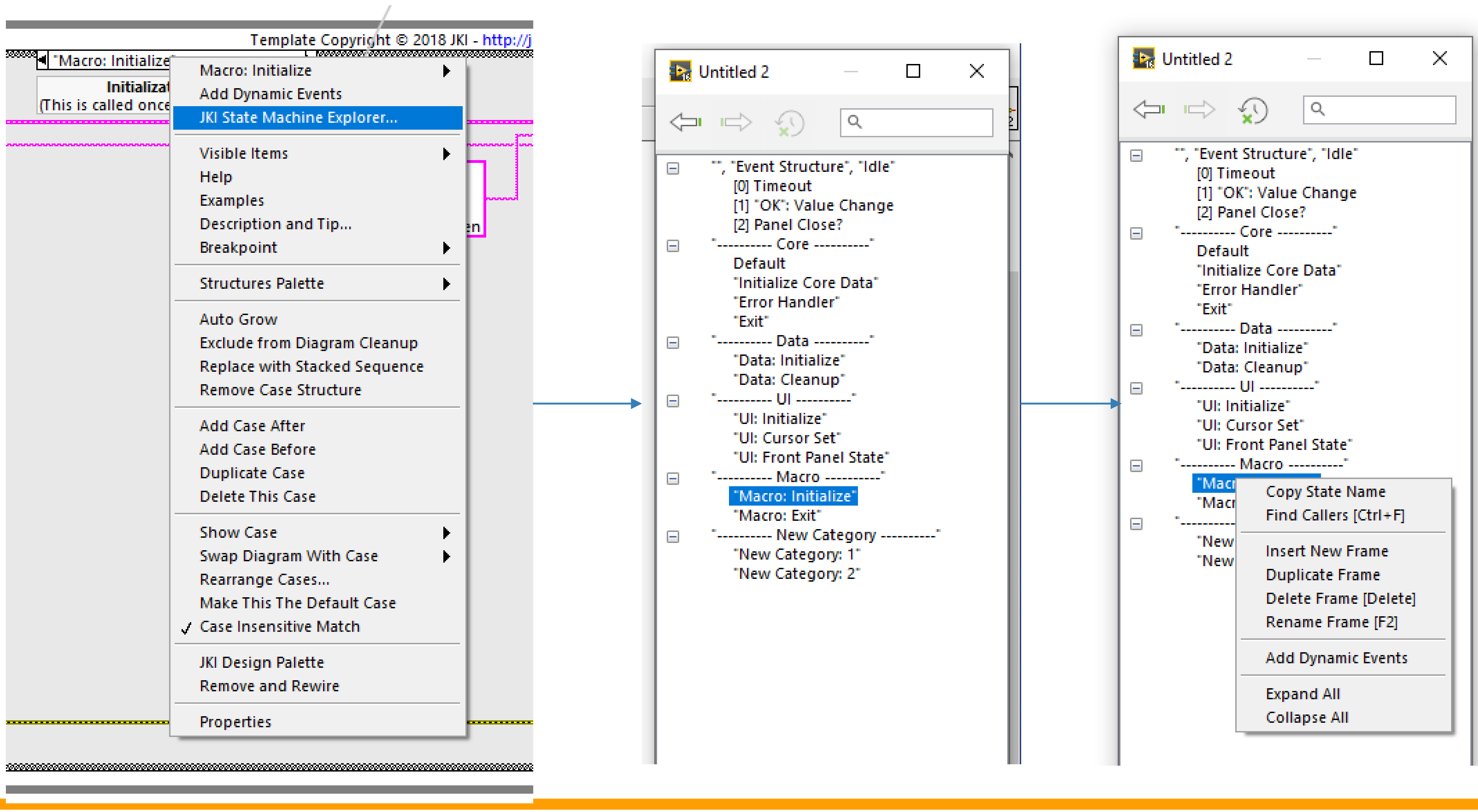


"Macro: Initialize"

Initialization Macro
(This is called once, when the VI starts)

```
Data: Initialize
Initialize Core Data
UI: Initialize
UI: Front Panel State >> Open
```

JKI State Machine Explorer



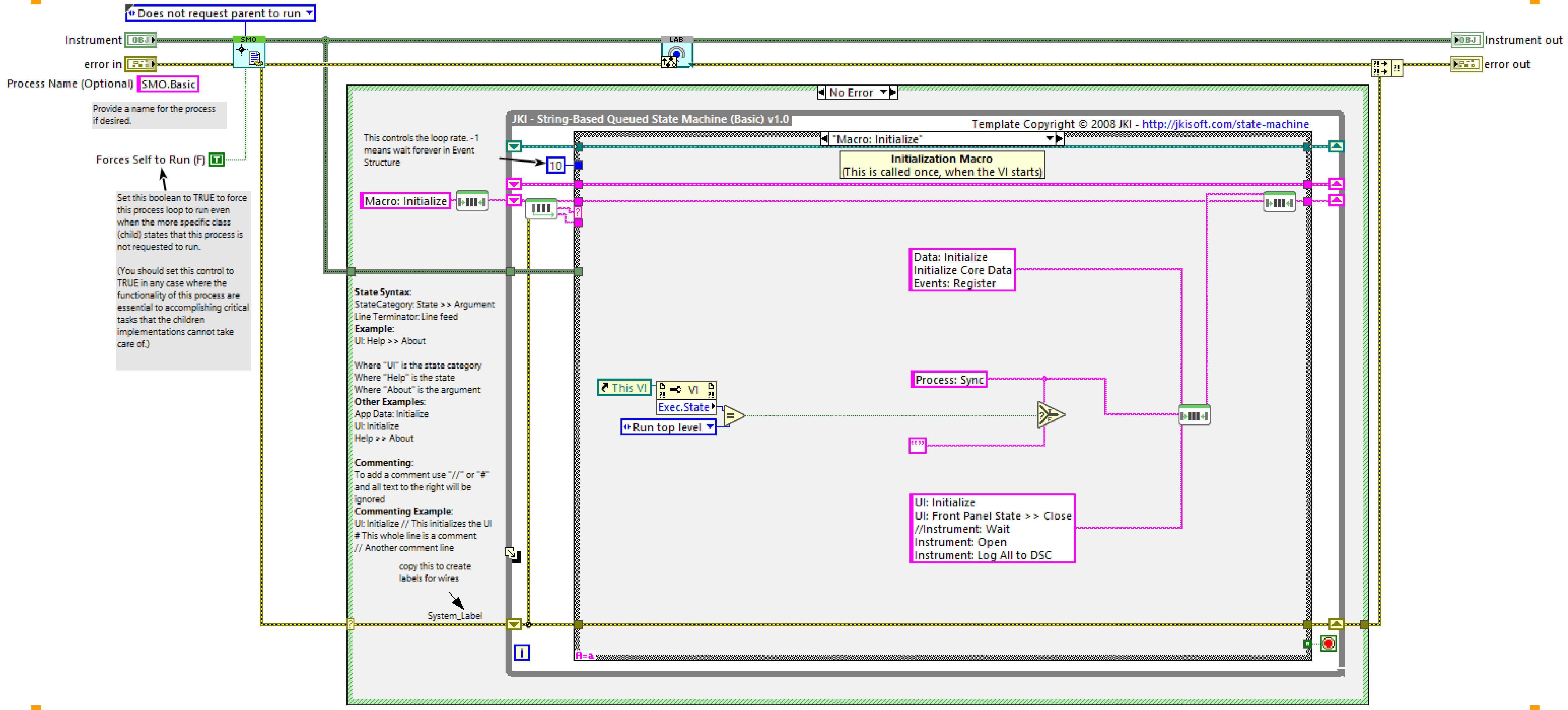
http://blog.jki.net/products/state-machine/jki-state-machine-best-practices

1. Don’t hide your state strings in subVIs
2. Don’t add code and logic inside the Event Structure
3. Keep the Original Size (i.e. don’t grow the structures)
4. Use macros instead of “chaining” together sequential states
5. Left-justify State Strings instead of Right-justify

https://www.youtube.com/watch?v=XJFujhIuZdU
https://www.youtube.com/watch?v=5H0lrLXZoq8

JKI State Machine *Objects*

State Machine Objects



Provide a name for the process if desired.

Set this boolean to TRUE to force this process loop to run even when the more specific class (child) states that this process is not requested to run.

(You should set this control to TRUE in any case where the functionality of this process are essential to accomplishing critical tasks that the children implementations cannot take care of.)

This controls the loop rate. -1 means wait forever in Event Structure

State Syntax:
StateCategory: State >> Argument
Line Terminator: Line feed
Example:
UI: Help >> About

Where "UI" is the state category
Where "Help" is the state
Where "About" is the argument
Other Examples:
App Data: Initialize
UI: Initialize
Help >> About

Commenting:
To add a comment use "//" or "#" and all text to the right will be ignored
Commenting Example:
UI: Initialize // This initializes the UI
This whole line is a comment
// Another comment line

copy this to create labels for wires

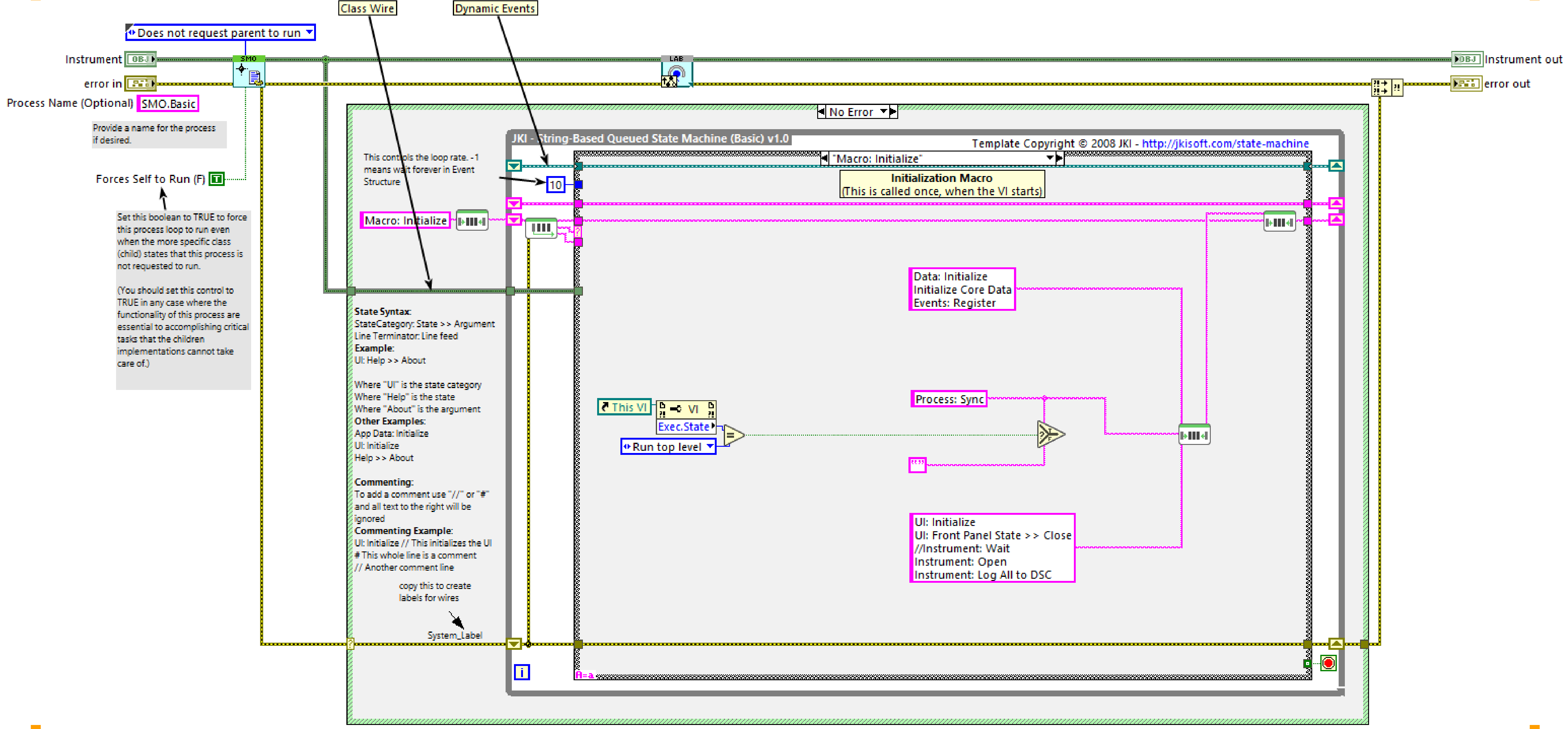
Initialization Macro
(This is called once, when the VI starts)

Data: Initialize
Initialize Core Data
Events: Register

Process: Sync

UI: Initialize
UI: Front Panel State >> Close
//Instrument: Wait
Instrument: Open
Instrument: Log All to DSC

State Machine Objects



Provide a name for the process if desired.

Set this boolean to TRUE to force this process loop to run even when the more specific class (child) states that this process is not requested to run.

(You should set this control to TRUE in any case where the functionality of this process are essential to accomplishing critical tasks that the children implementations cannot take care of.)

This controls the loop rate. -1 means wait forever in Event Structure

State Syntax:
StateCategory: State >> Argument
Line Terminator: Line feed
Example:
UI: Help >> About

Where "UI" is the state category
Where "Help" is the state
Where "About" is the argument
Other Examples:
App Data: Initialize
UI: Initialize
Help >> About

Commenting:
To add a comment use "//" or "#" and all text to the right will be ignored
Commenting Example:
UI: Initialize // This initializes the UI
This whole line is a comment
// Another comment line

copy this to create labels for wires

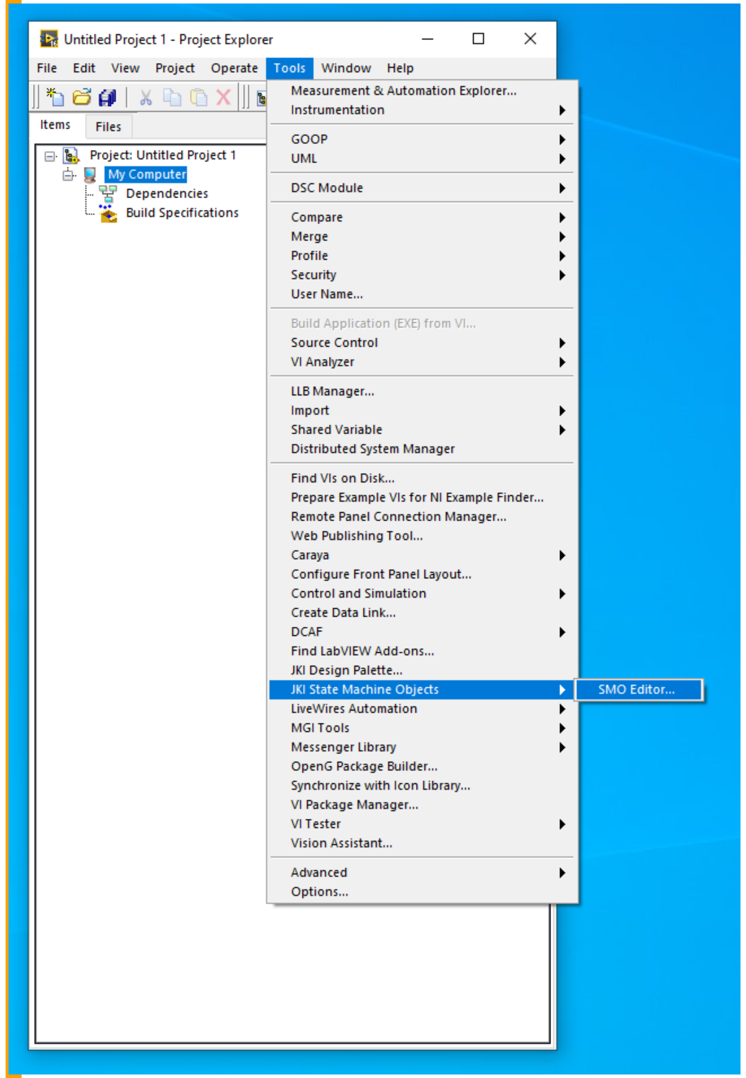
Initialization Macro
(This is called once, when the VI starts)

```
Data: Initialize
Initialize Core Data
Events: Register
```

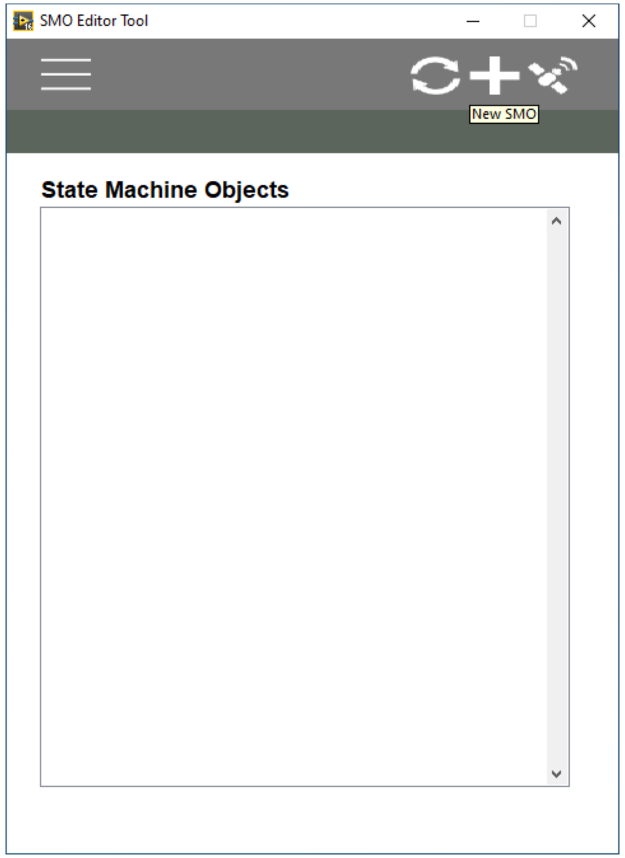
```
Process: Sync
```

```
UI: Initialize
UI: Front Panel State >> Close
//Instrument: Wait
Instrument: Open
Instrument: Log All to DSC
```

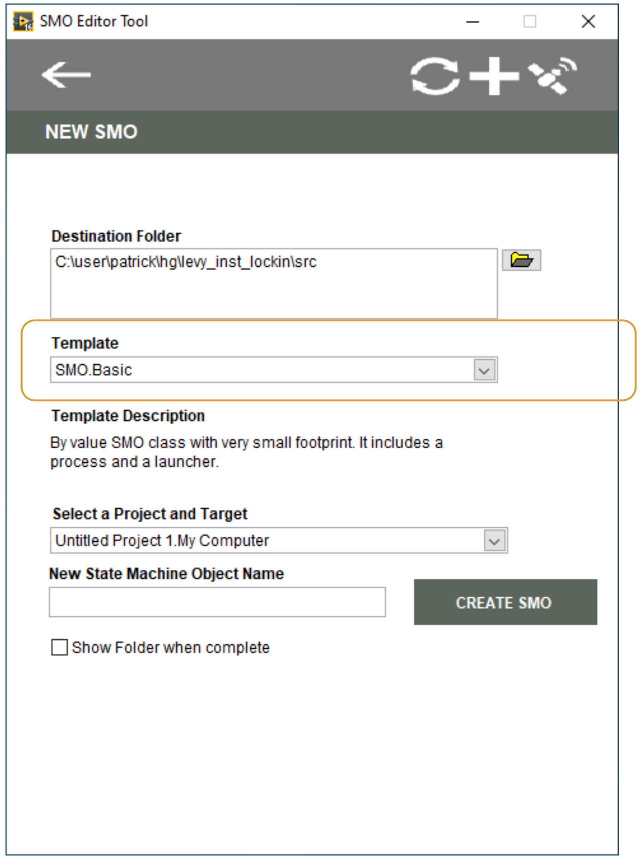
Awesome! How do I make an SMO?



1. Open the SMO Editor



2. Click "New SMO"



Destination Folder: C:\user\patrick\hg\levy_inst_lockin\src

Template: SMO.Basic

Template Description: By value SMO class with very small footprint. It includes a process and a launcher.

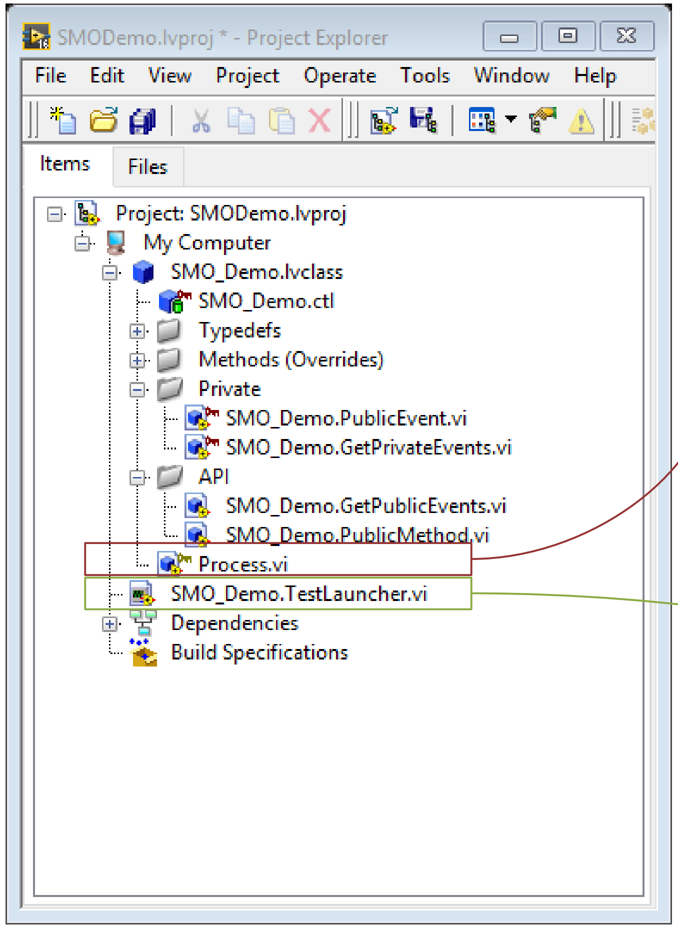
Select a Project and Target: Untitled Project 1.My Computer

New State Machine Object Name:

☐ Show Folder when complete

3. Choose SMO Type

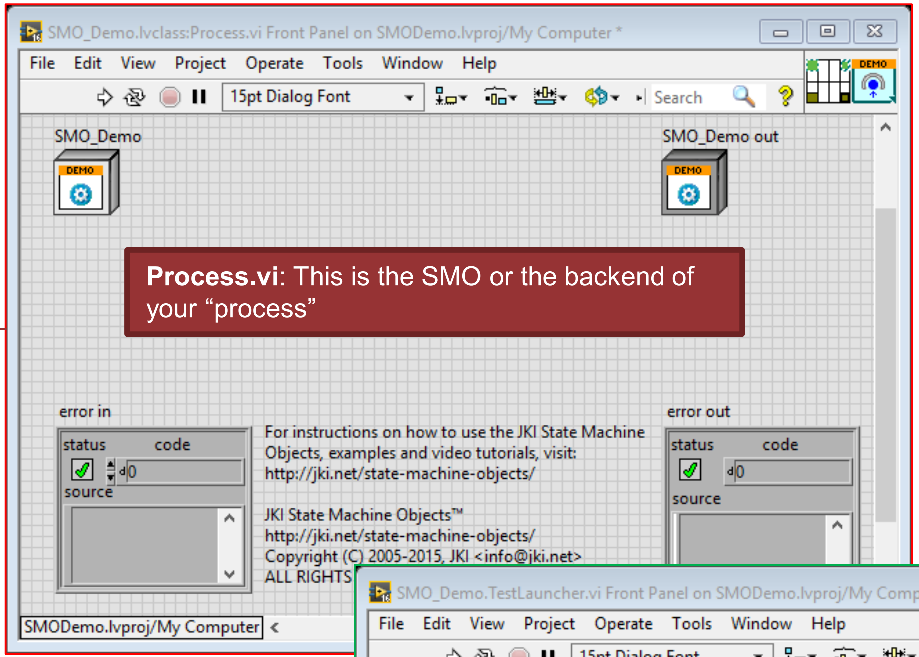
SMO Intro 1


SMODemo.lvproj * - Project Explorer

File Edit View Project Operate Tools Window Help

Items Files

```
Project: SMODemo.lvproj
  My Computer
    SMO_Demo.lvclass
      SMO_Demo.ctl
      Typedefs
      Methods (Overrides)
      Private
        SMO_Demo.PublicEvent.vi
        SMO_Demo.GetPrivateEvents.vi
      API
        SMO_Demo.GetPublicEvents.vi
        SMO_Demo.PublicMethod.vi
      Process.vi
    SMO_Demo.TestLauncher.vi
    Dependencies
    Build Specifications
```


SMO_Demo.lvclass:Process.vi Front Panel on SMODemo.lvproj/My Computer *

File Edit View Project Operate Tools Window Help

SMO_Demo — SMO_Demo out

error in — status, code 0, source

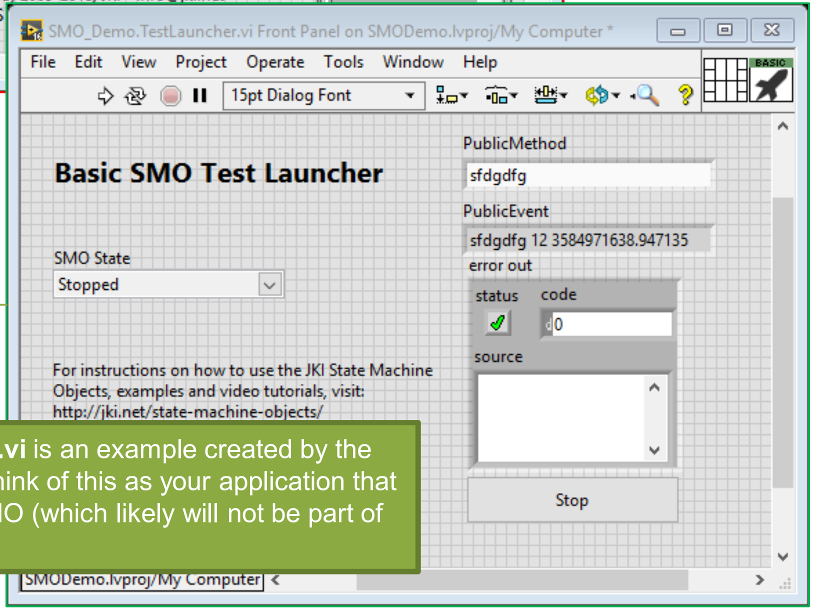
error out — status, code 0, source

For instructions on how to use the JKI State Machine Objects, examples and video tutorials, visit: http://jki.net/state-machine-objects/

JKI State Machine Objects™
http://jki.net/state-machine-objects/
Copyright (C) 2005-2015, JKI <info@jki.net>
ALL RIGHTS

SMODemo.lvproj/My Computer

Process.vi: This is the SMO or the backend of your “process”


SMO_Demo.TestLauncher.vi Front Panel on SMODemo.lvproj/My Computer *

File Edit View Project Operate Tools Window Help

Basic SMO Test Launcher

SMO State: Stopped

PublicMethod: sfdgdfg

PublicEvent: sfdgdfg 12 3584971638.947135

error out — status, code 0, source

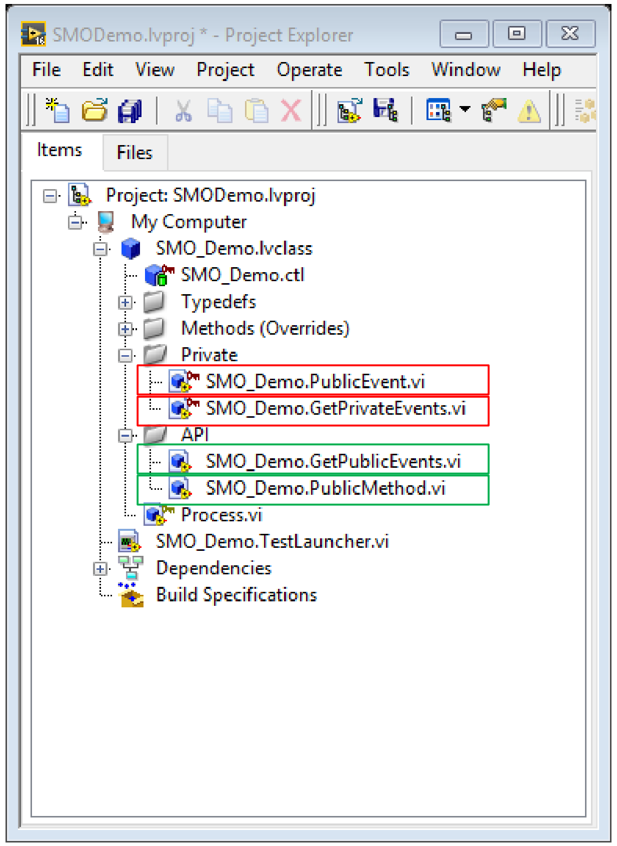
Stop

For instructions on how to use the JKI State Machine Objects, examples and video tutorials, visit: http://jki.net/state-machine-objects/

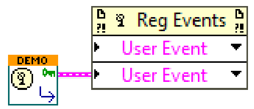
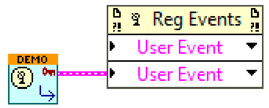
SMODemo.lvproj/My Computer

TestLauncher.vi is an example created by the SMO Editor. Think of this as your application that is *using* the SMO (which likely will not be part of your class.)

SMO Intro 2a

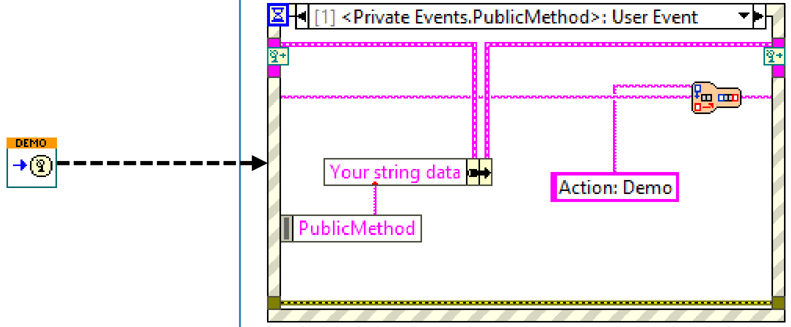


```
SMODemo.lvproj * - Project Explorer
File  Edit  View  Project  Operate  Tools  Window  Help
Items  Files
Project: SMODemo.lvproj
  My Computer
    SMO_Demo.lvclass
      SMO_Demo.ctl
      Typedefs
      Methods (Overrides)
      Private
        SMO_Demo.PublicEvent.vi
        SMO_Demo.GetPrivateEvents.vi
      API
        SMO_Demo.GetPublicEvents.vi
        SMO_Demo.PublicMethod.vi
      Process.vi
    SMO_Demo.TestLauncher.vi
    Dependencies
    Build Specifications
```

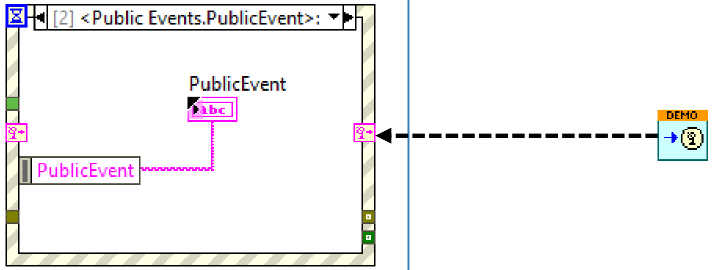
	Test Launcher	Process.vi
Startup:	 Register for public events	 Register for private events

During Execution:

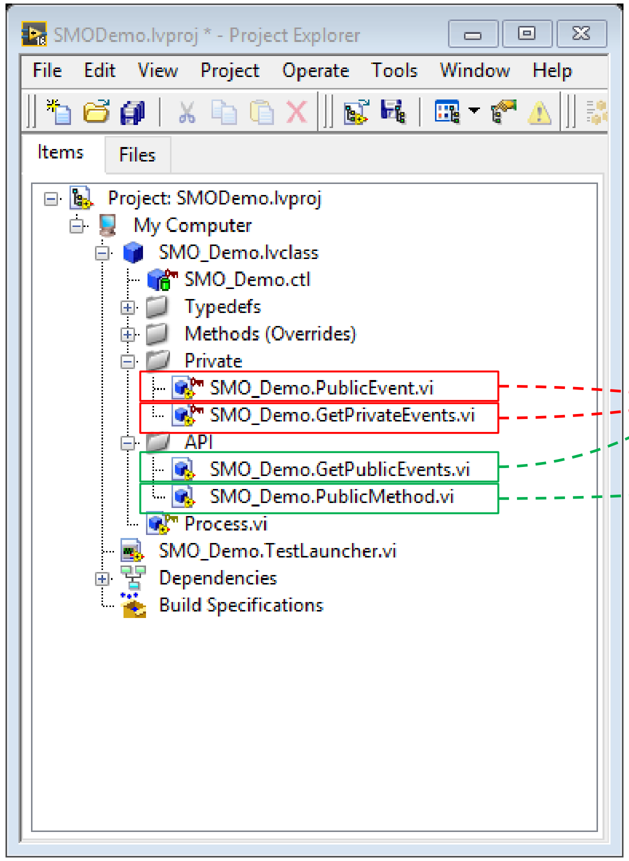
Calling PublicMethod from TestLauncher triggers event and sends data to process.vi



Calling PublicEvent from Process.vi triggers event and sends data to TestLauncher



SMO Intro 2b

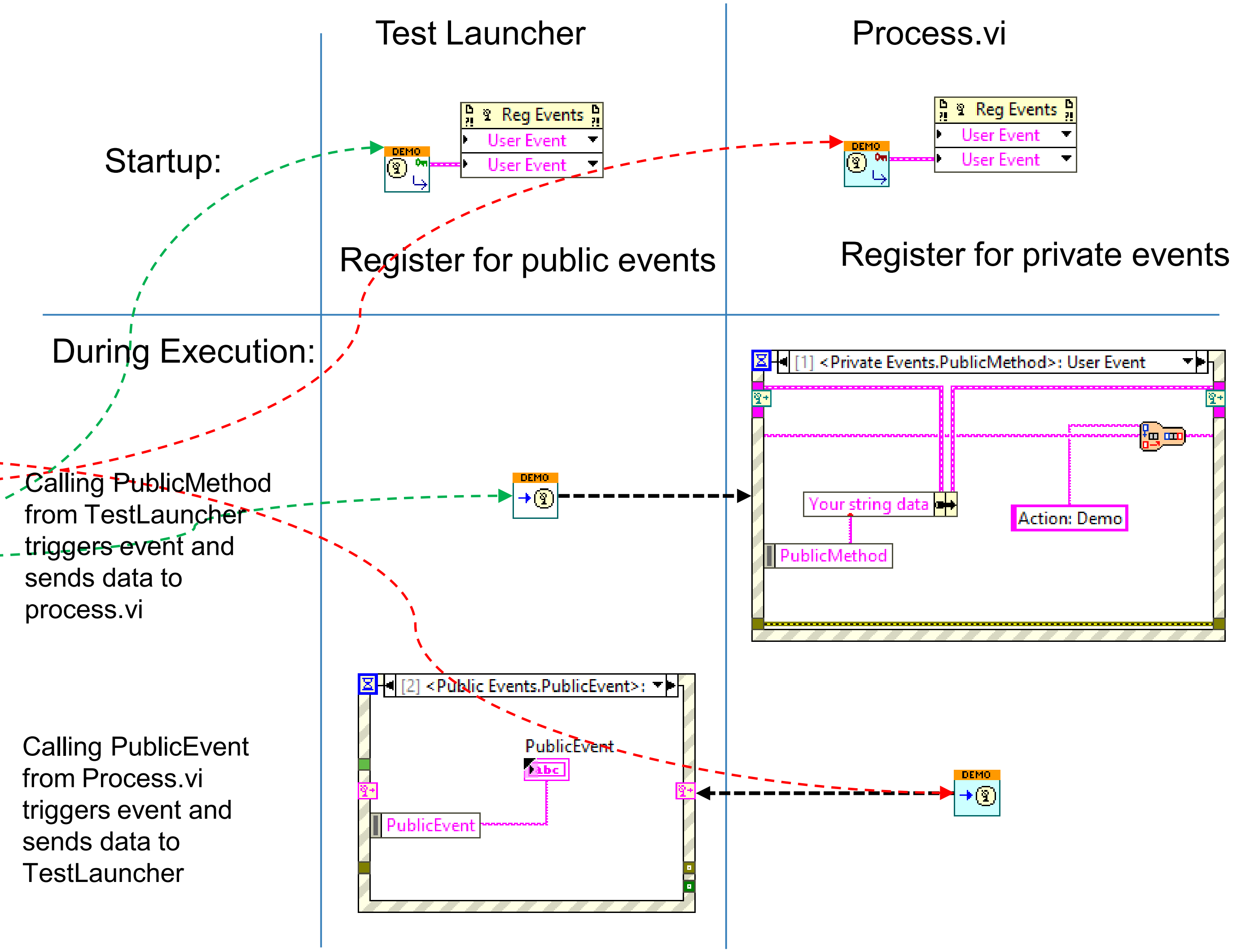


SMODemo.lvproj * - Project Explorer

File Edit View Project Operate Tools Window Help

Items Files

- Project: SMODemo.lvproj
  - My Computer
    - SMO_Demo.lvclass
      - SMO_Demo.ctl
      - Typedefs
      - Methods (Overrides)
      - Private
        - SMO_Demo.PublicEvent.vi
        - SMO_Demo.GetPrivateEvents.vi
      - API
        - SMO_Demo.GetPublicEvents.vi
        - SMO_Demo.PublicMethod.vi
      - Process.vi
    - SMO_Demo.TestLauncher.vi
    - Dependencies
    - Build Specifications



Startup:

Register for public events

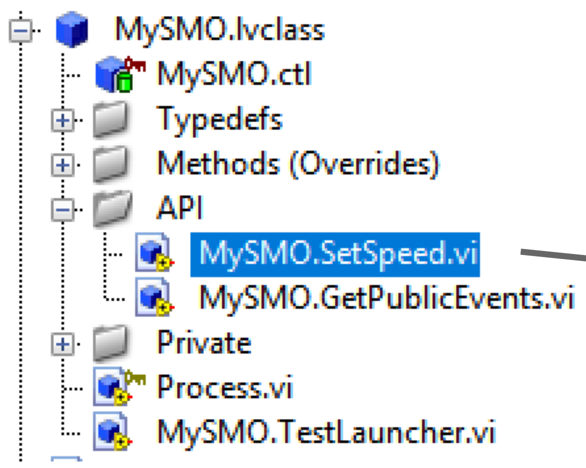
Register for private events

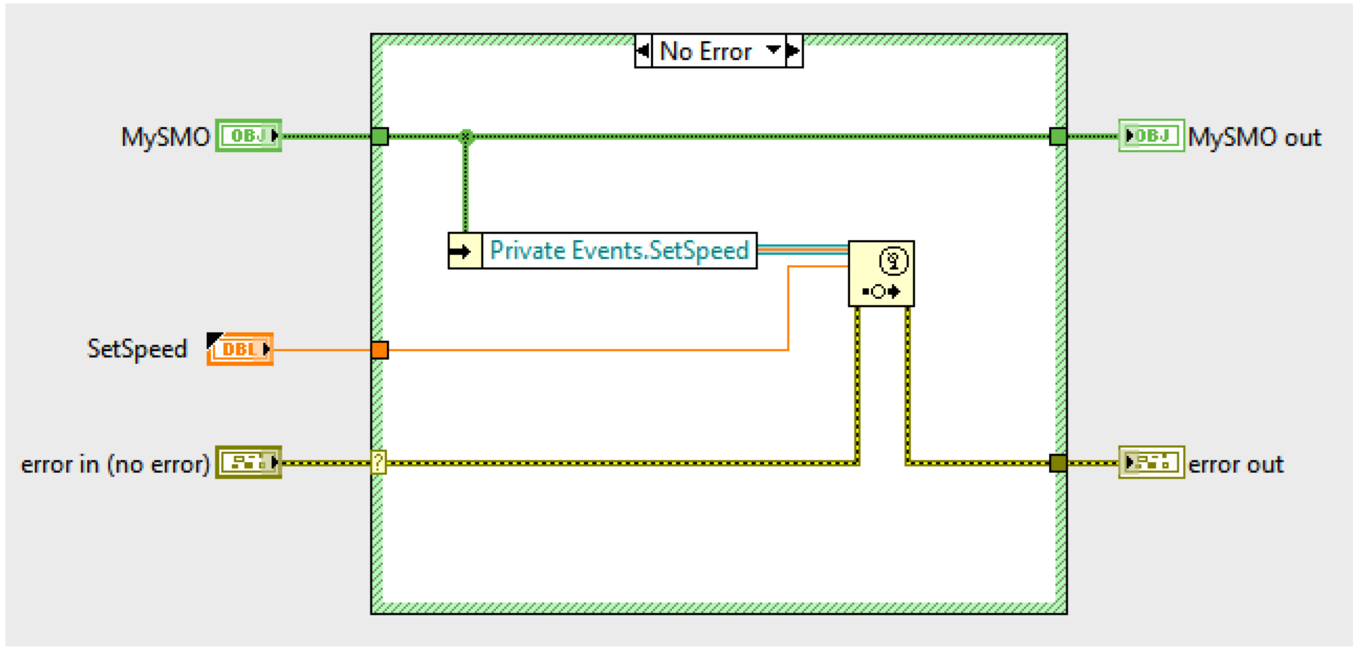
During Execution:

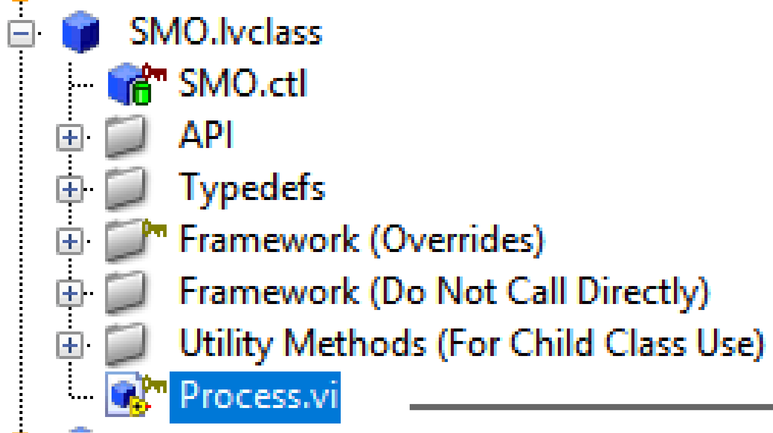
Calling PublicMethod from TestLauncher triggers event and sends data to process.vi

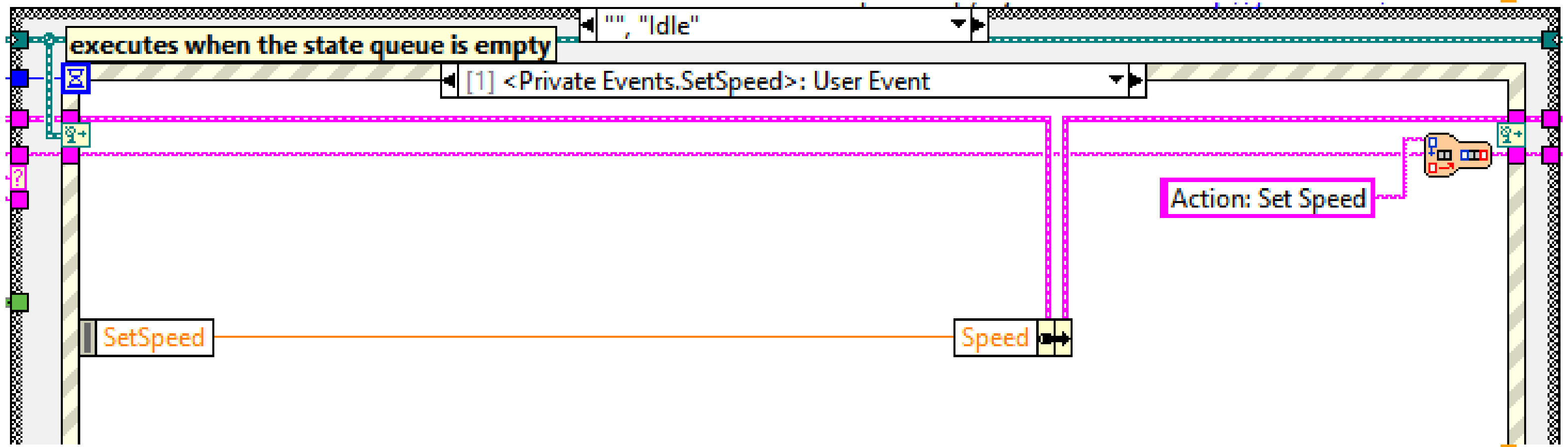
Calling PublicEvent from Process.vi triggers event and sends data to TestLauncher

State Machine Objects: Dynamic Events

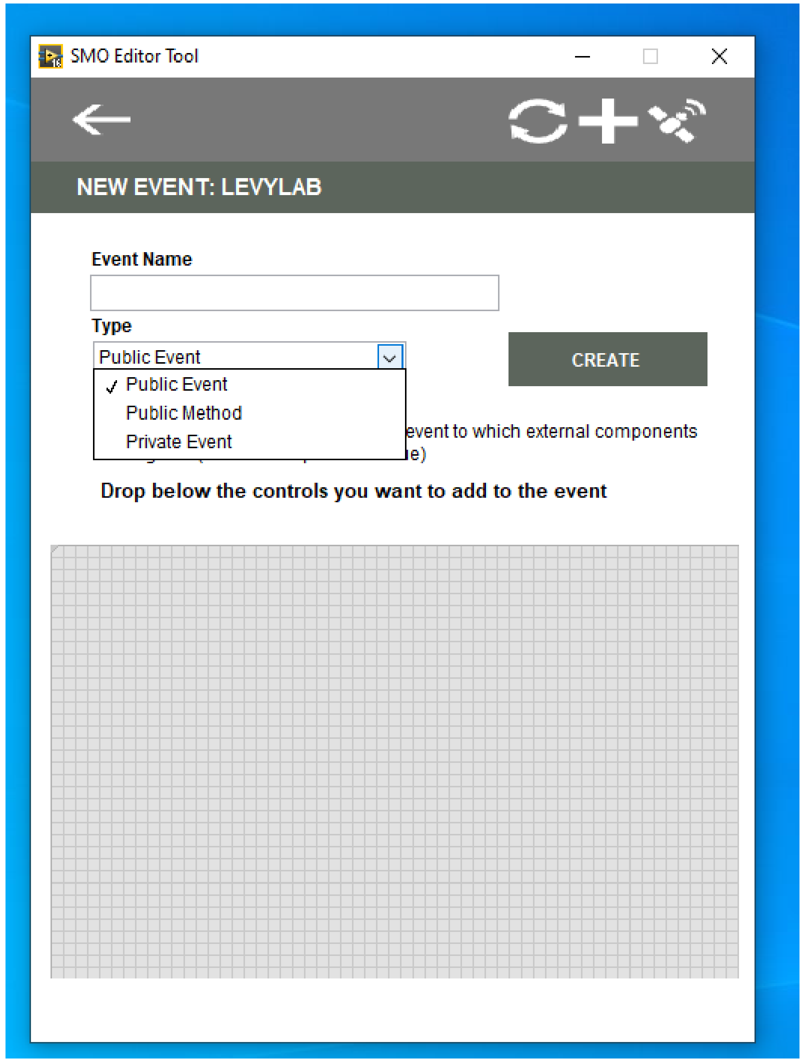

MySMO.lvclass
MySMO.ctl
Typedefs
Methods (Overrides)
API
MySMO.SetSpeed.vi
MySMO.GetPublicEvents.vi
Private
Process.vi
MySMO.TestLauncher.vi




SMO.lvclass
SMO.ctl
API
Typedefs
Framework (Overrides)
Framework (Do Not Call Directly)
Utility Methods (For Child Class Use)
Process.vi


"", "Idle"
executes when the state queue is empty
[1] <Private Events.SetSpeed>: User Event
Action: Set Speed
SetSpeed
Speed

Use SMO Editor to create Public Events and Public Methods



SMO Editor Tool

NEW EVENT: LEVYLAB

Event Name

Type

Public Event

✓ Public Event
Public Method
Private Event

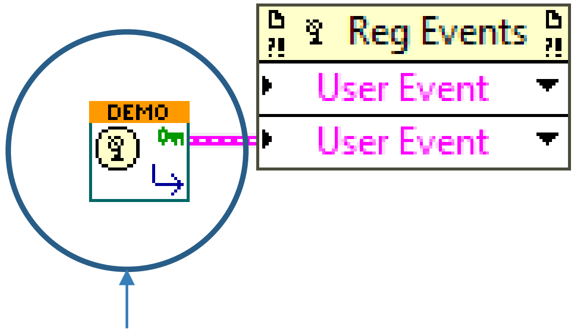
CREATE

event to which external components
e)

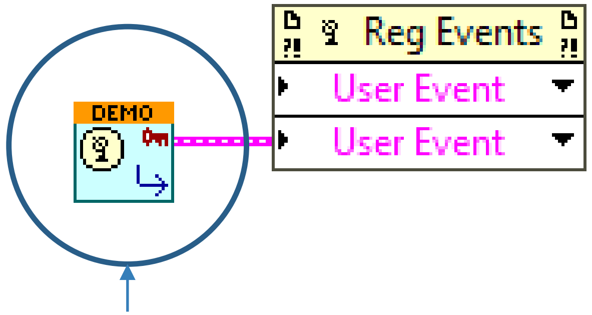
Drop below the controls you want to add to the event

SMO Intro 3

Change your icons to make them useful!

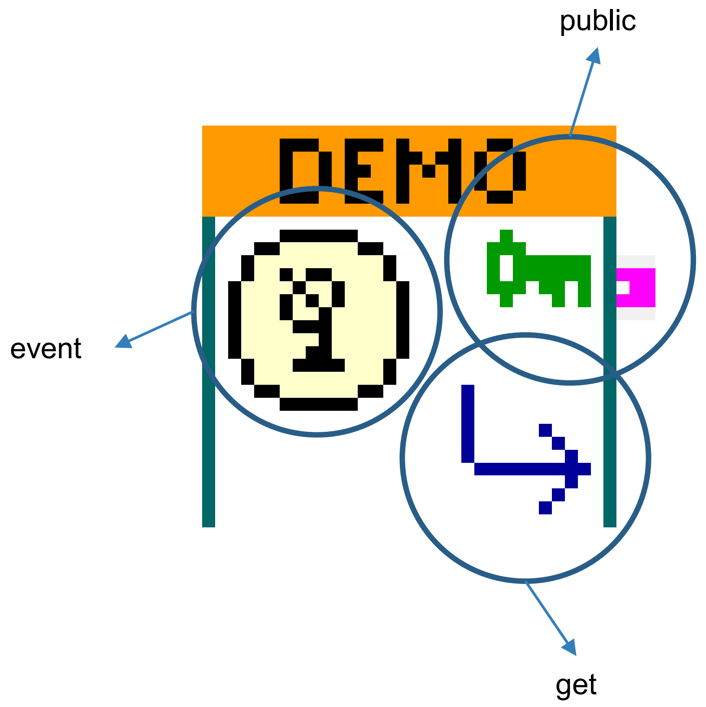


White background = public method



Blue background = private method

Use glyphs provided by LabVIEW to make your life easy!



public
DEMO
event
get