

# The Ins and Outs of **X**Controls

Jeffrey Habets  
NI Certified LabVIEW Architect  
[www.vi-tech.nl](http://www.vi-tech.nl)



# Agenda

- Introduction
- Anatomy of an Xcontrol
- Dissecting the facade: Application ↔ XControl interaction
- State Persistence
- Version Control
- Properties and Methods
- Debugging
- Programming Techniques



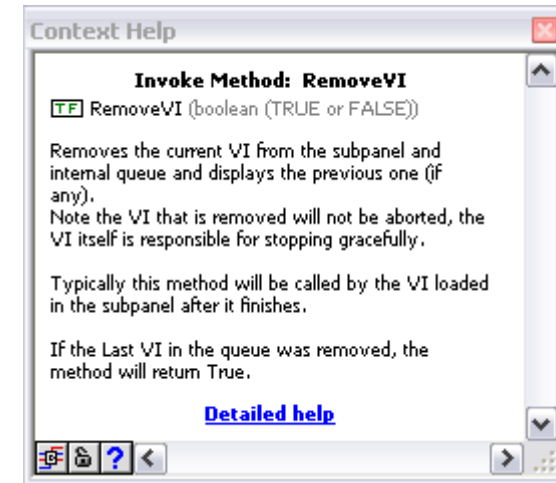
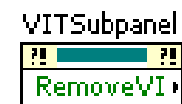
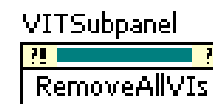
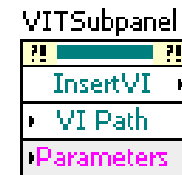
## Introduction

- An XControl has custom behaviour in addition to custom appearance
- Encapsulates UI code into abstract component
  - Reduces complexity and increases readability of main block diagram
- Creates reusable and distributable UI components
- Since LabVIEW 8.0
- Need LV Professional Development suite to create

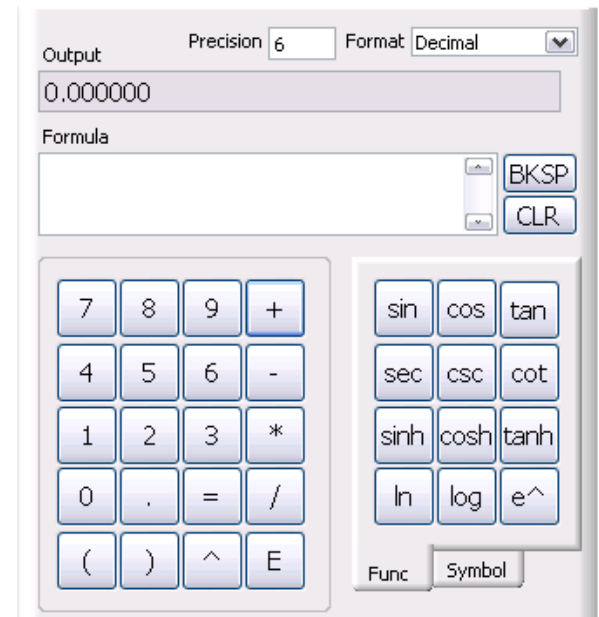


# Introduction – Typical use

- Add functionality to existing controls
  - New properties or methods
  - Analysis / processing
- Combine existing controls
  - Combine multiple controls into a single data type, abstracting out code and information related to the display of the controls

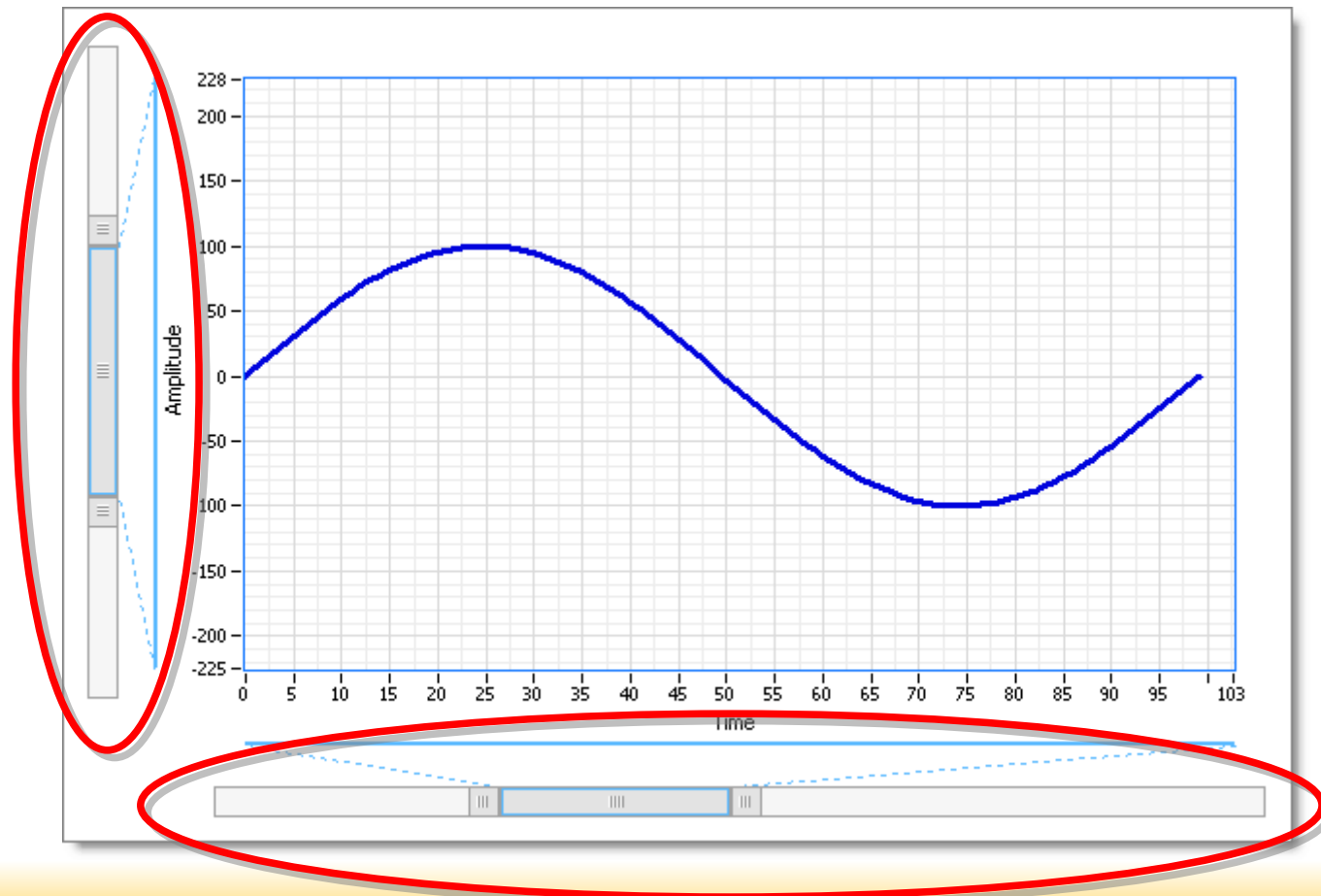


Calculator XControl



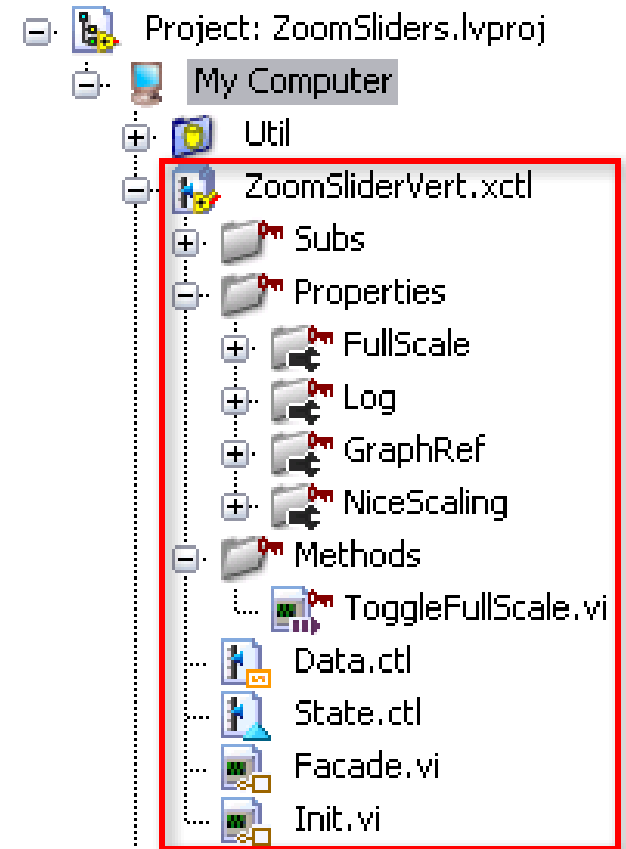
## Introduction – Typical use

- Create new controls
  - Use the picture control
  - Custom graphs are a common example (e.g. new graphs in LV2009)



# Anatomy of an XControl

- Xcontrol library (.xctl)
  - Abilities
  - Properties
  - Methods
  - Dependent files
- Create
  - My Computer → New → XControl
  - File → New... → select XControl template



## Anatomy of an XControl - Abilities

- Data (required)
  - Typedef that defines the XControl's datatype
- State a.k.a. Display State (required)
  - Typedef that specifies any information other than the XControl's data that determines the appearance of the control
- Facade (required)
  - VI of which the front panel defines the appearance of the control and the block diagram defines its behaviour



## Anatomy of an XControl - Abilities

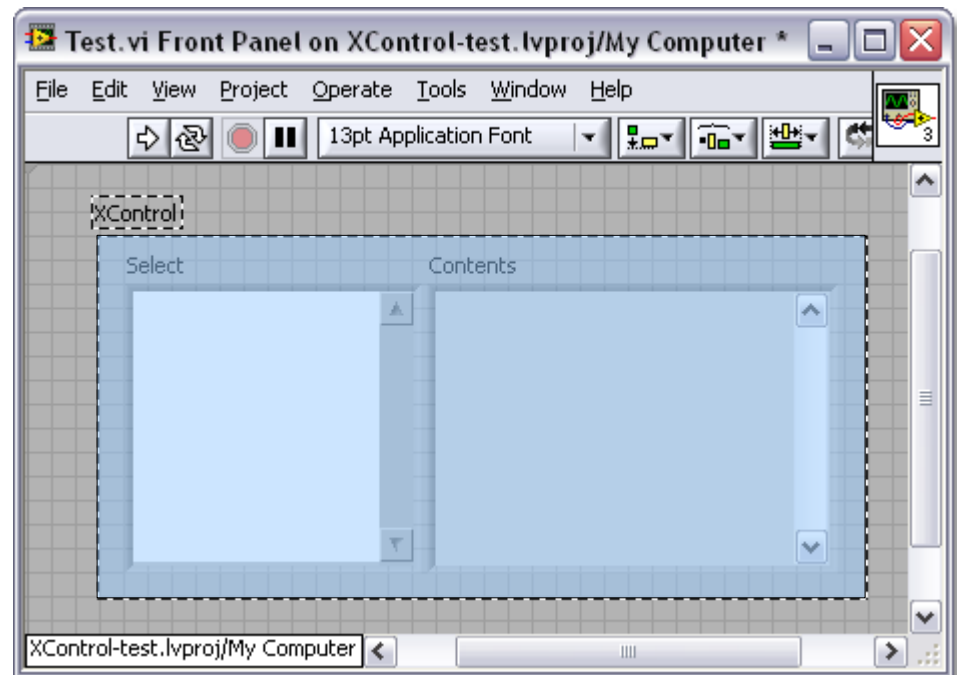
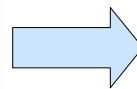
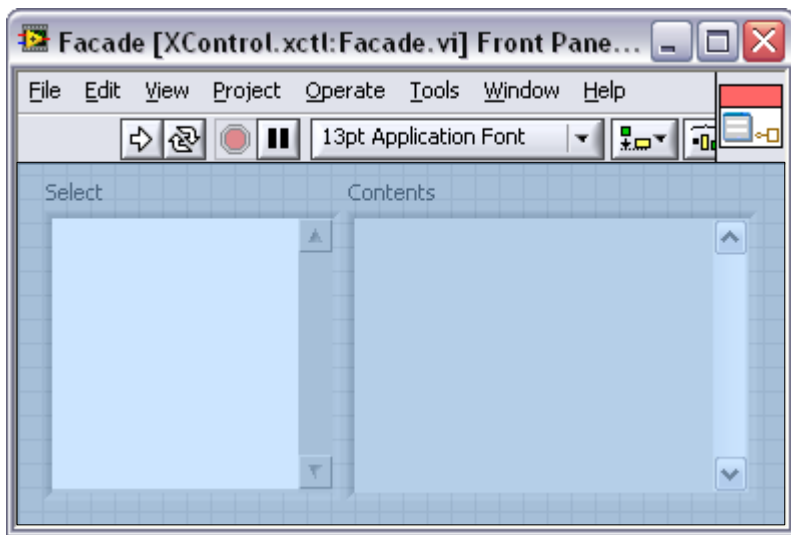
- Init (required)
  - VI called by LV when the control is first dropped on a front panel or when a VI containing the control is loaded into memory. Initializes the display state and/or allocate resources used by the XControl
- Uninit (optional)
  - VI called by LV when XControl is removed from memory. Release resources allocated by the XControl
- Convert State For Save (optional)
  - VI called by LV prior to saving the XControl. Use to clear non-persistent fields from the display state data (to reduce footprint on disk)





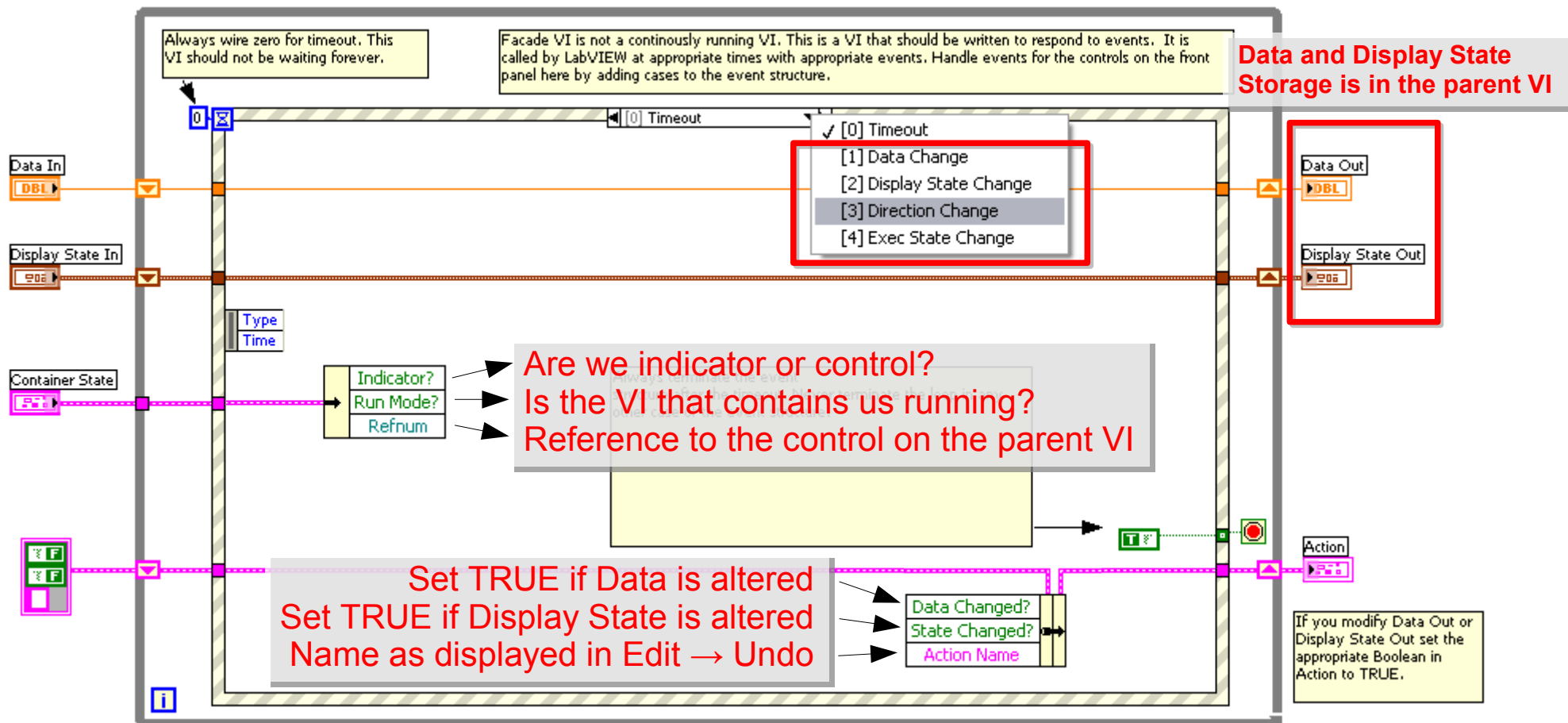
## Dissecting the Facade: Application ↔ XControl interaction

- Facade is the heart of the XControl
- The Facade is a not continuously running event handler VI
- Visible part of the front panel window defines the appearance of the XControl



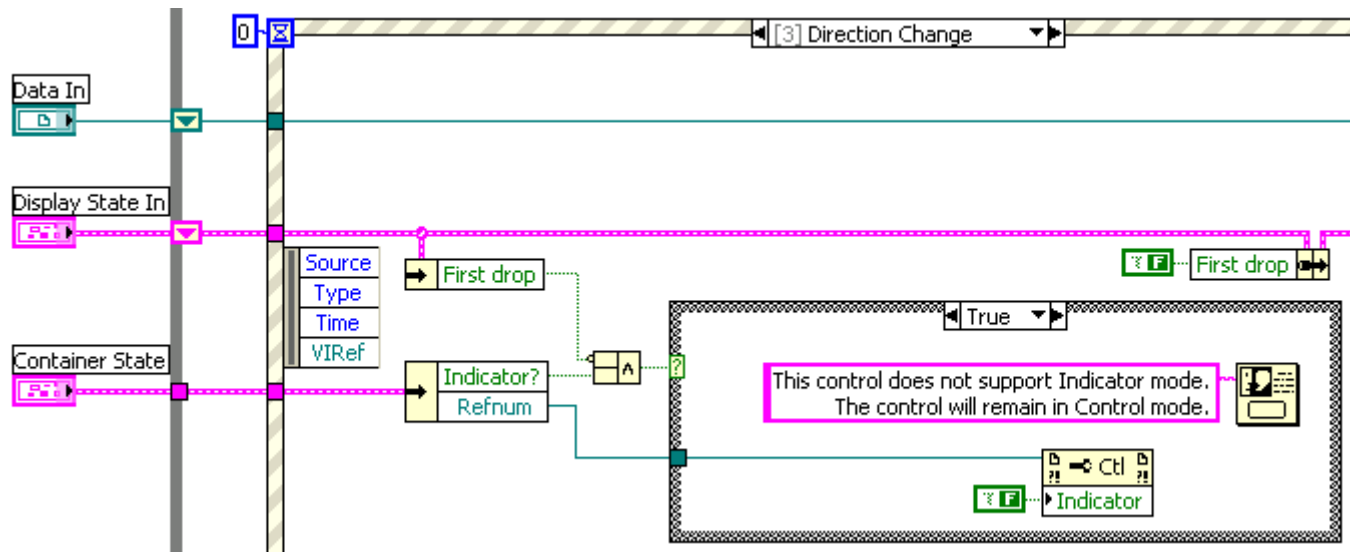
# Dissecting the Facade: Application ↔ XControl interaction

- VI that uses the control calls the Facade when events are available
- Most of the action happens here



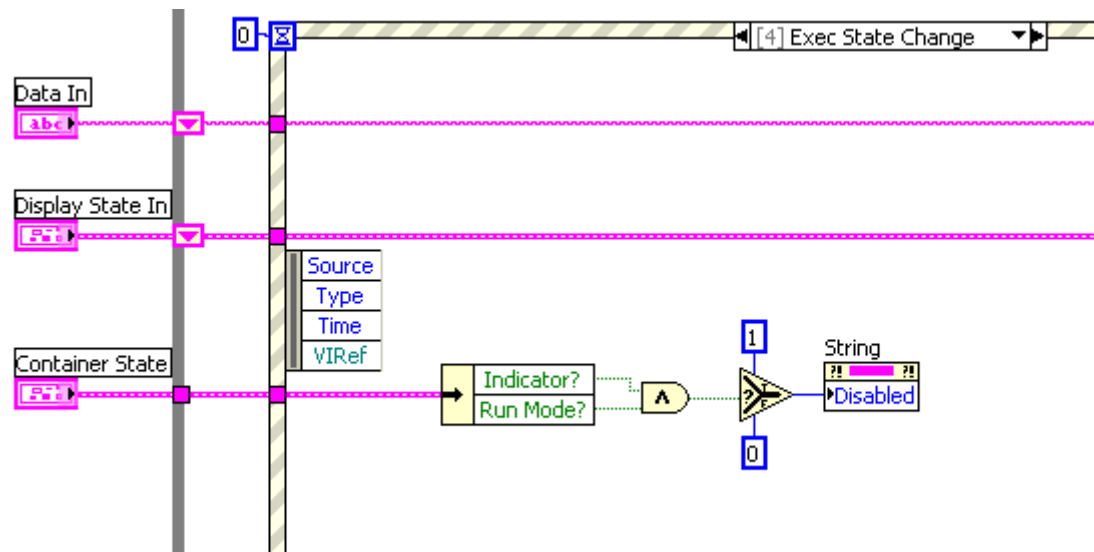
## Dissecting the Facade: **Direction Change** event

- Generated when an instance of the XControl changes from a control to an indicator or vice versa
- Update the appearance of the control here, e.g. using color, disabled state and visibility properties of controls that make up your XControl
- If applicable, force control to a certain mode



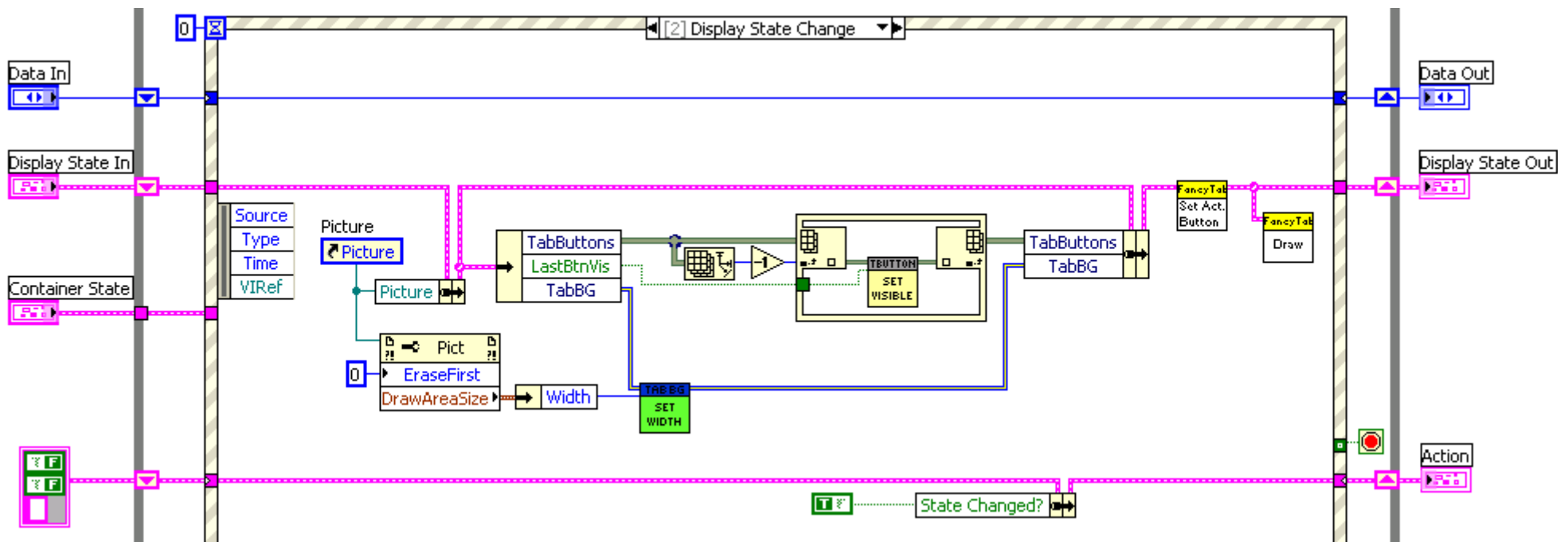
## Dissecting the Facade: Exec State Change event

- Generated when the parent VI changes from edit to run mode or vice versa
- Change appearance of control here if necessary
- Use Container State.Run Mode? To get current execution state



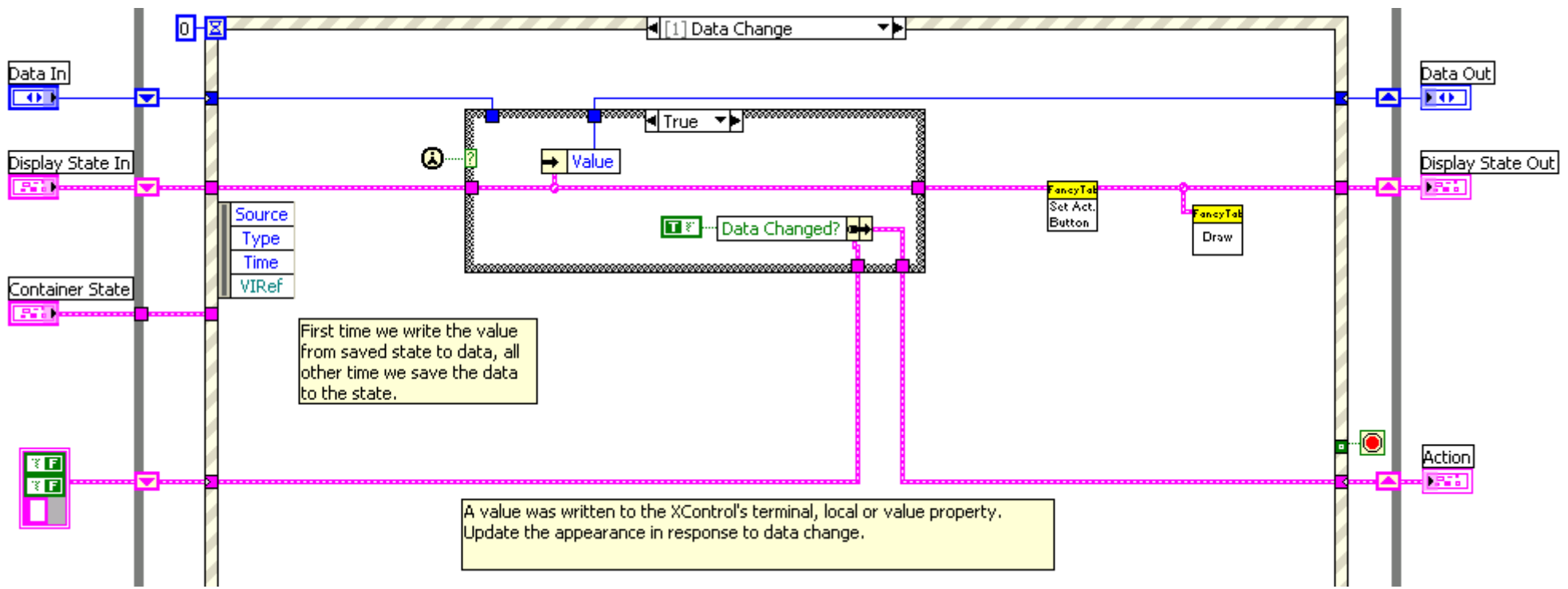
## Dissecting the Facade: Display State Change event

- Generated when a method or property of the control is executed
- Use the value in the Display State to update your control's appearance



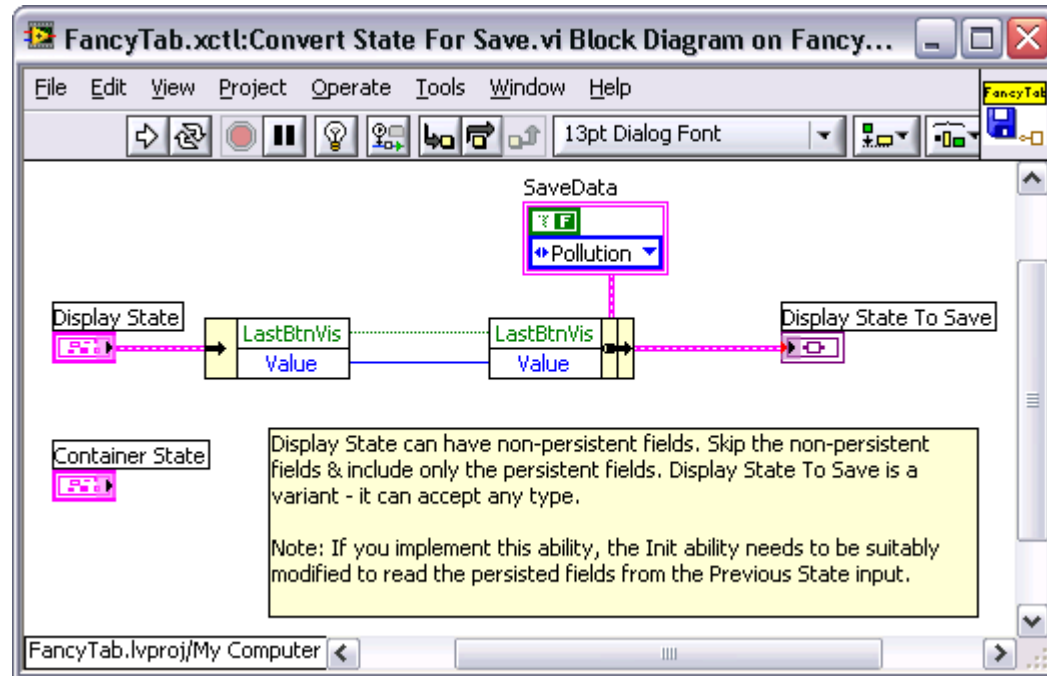
## Dissecting the Facade: Data Change event

- Generated when a value is written to the control's terminal, a local variable or a value property
- Update state and appearance as necessary



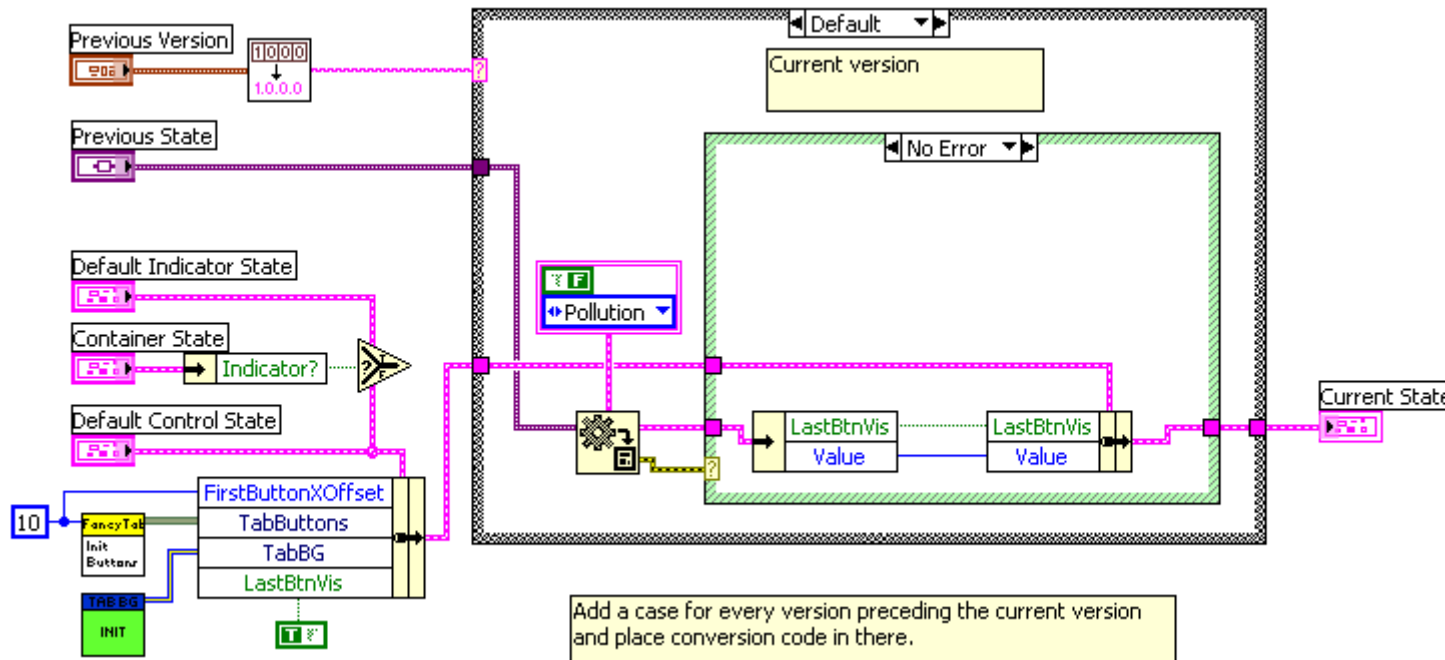
## State Persistence

- State of the XControl is stored in the parent VI
- Saved to disk when the parent VI is saved
- Use **Convert State For Save** ability to remove fields that don't need to be saved to disk (e.g. refnums)



# State Persistence

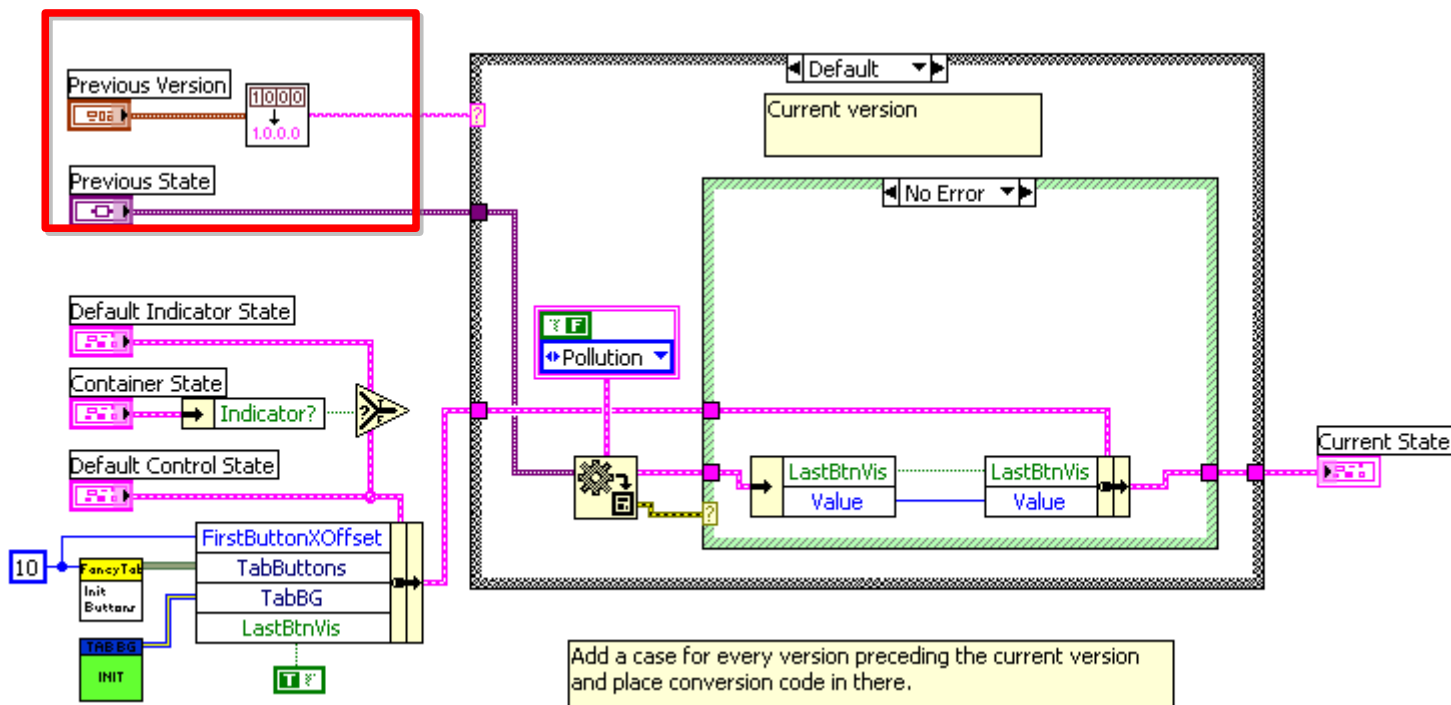
- Must implement state conversion code in Init ability to convert back the Saved State to Display State





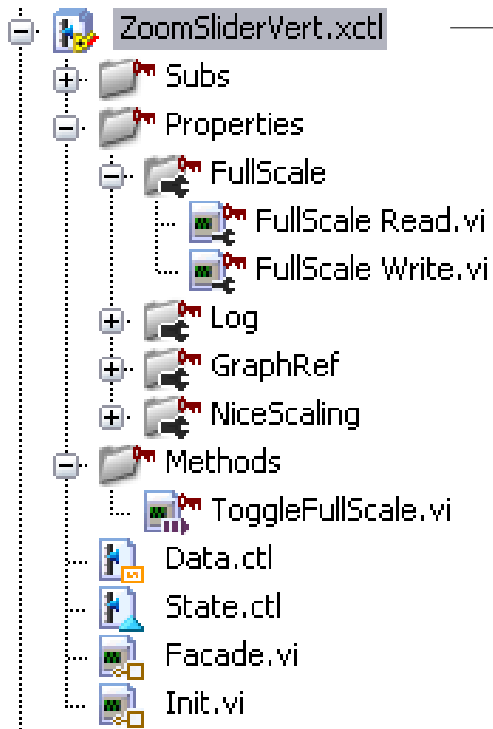
# Version Control

- When releasing new version of your control, use the Init ability to convert the State of previous versions of the control to the current State
- The version number can be set in the XControl properties dialog
- Save old State data type under new name (linked to version)

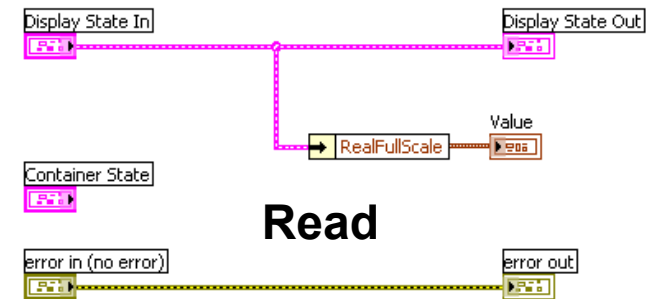
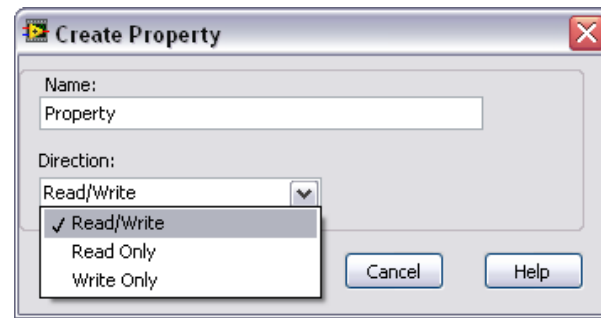


# Properties and Methods

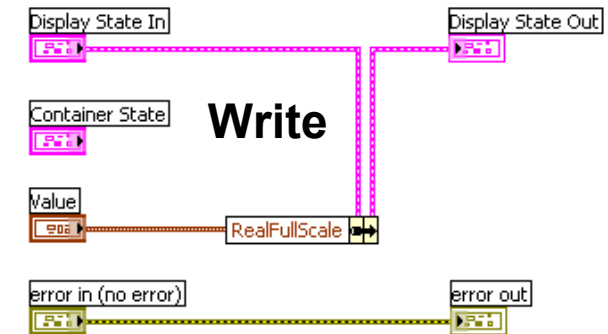
- Properties and methods basically just read or update the Display State
- Property exists of 1 or 2 VI's



New → Property



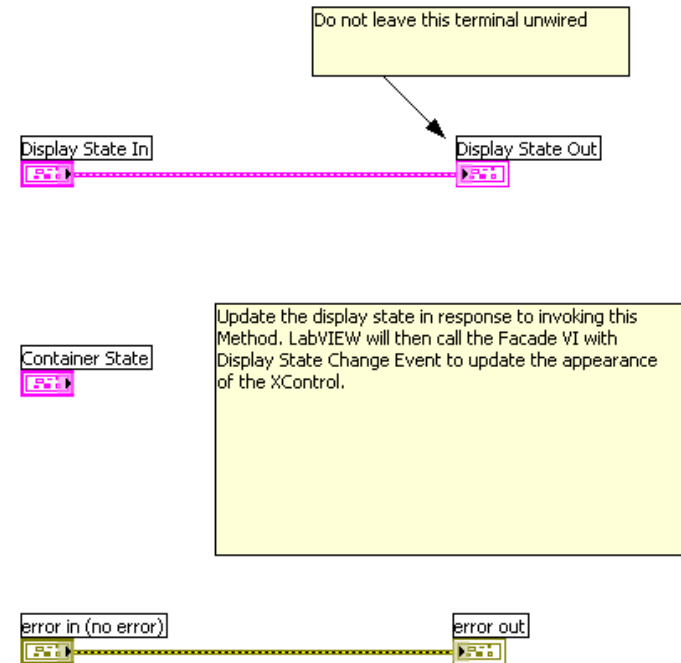
Read



Write

## Properties and Methods

- Methods have basically the same template as properties
- You can add multiple inputs/outputs or have none at all

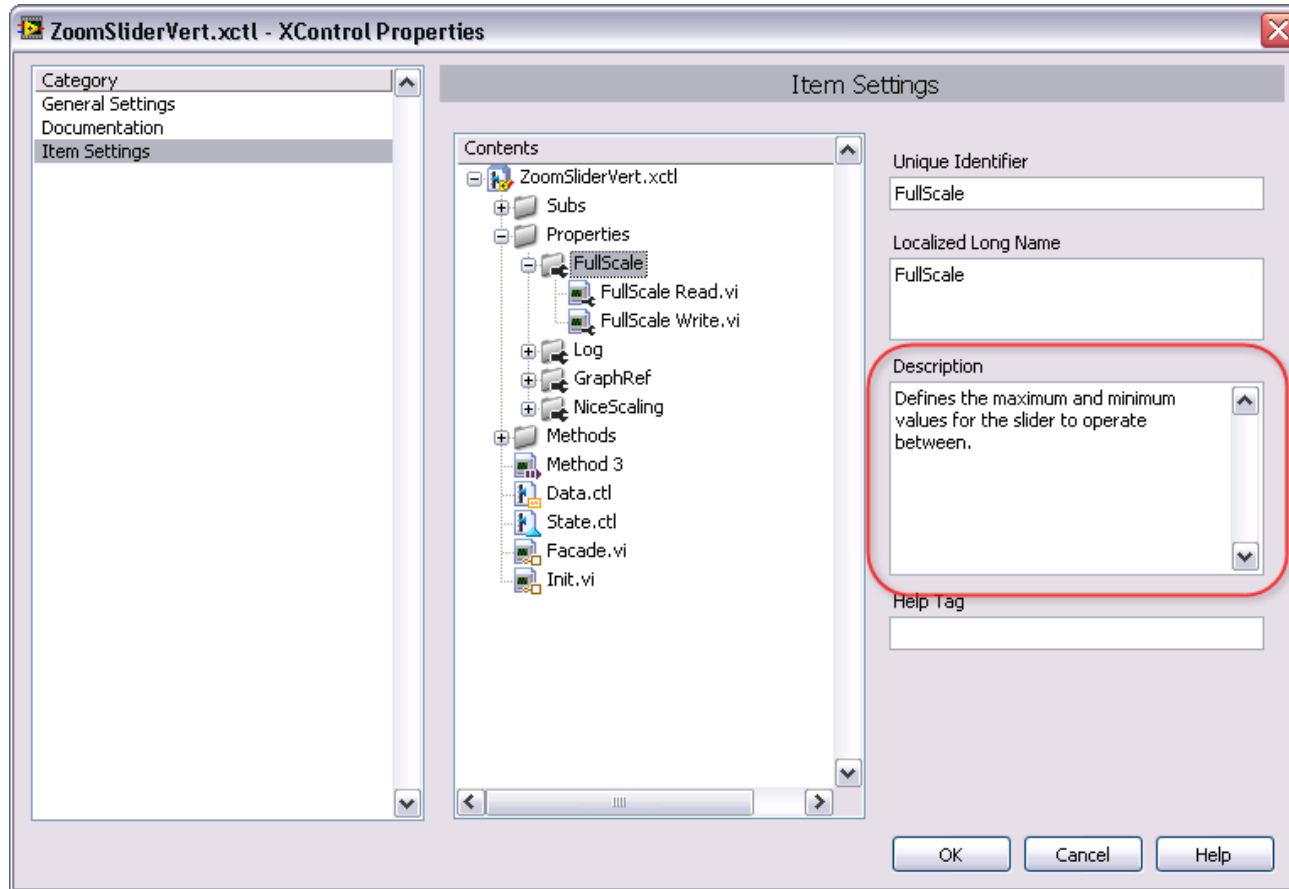


- Default methods and properties can't be replaced, removed or overridden
- Can't inherit methods or properties from controls on the Facade



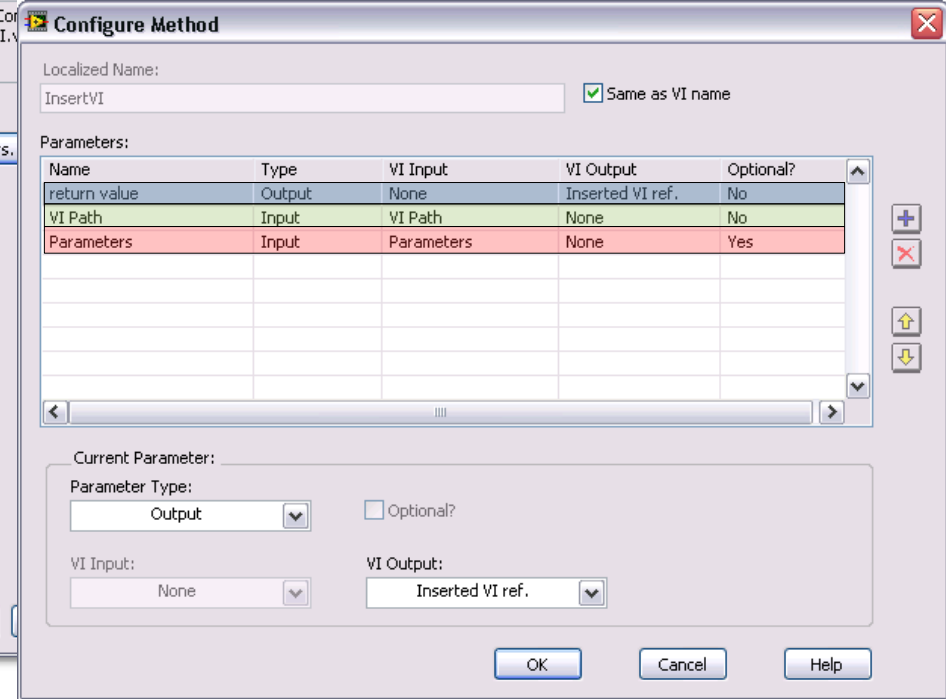
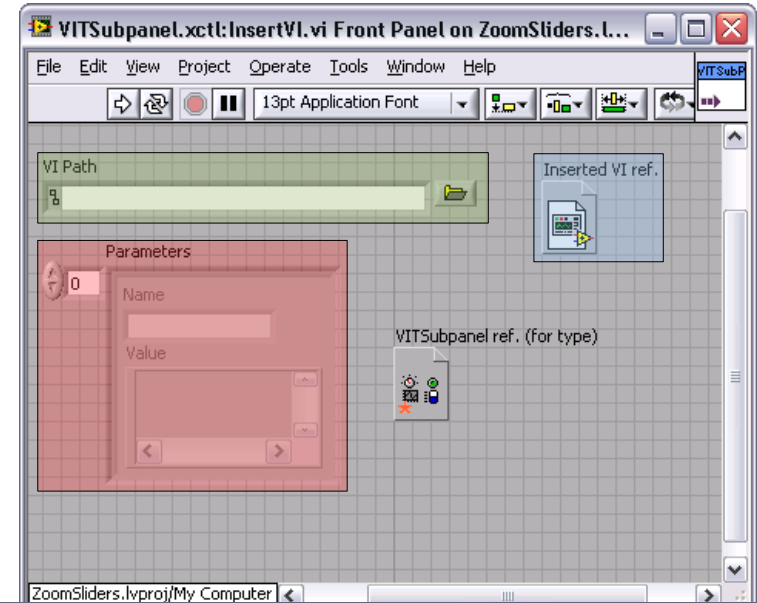
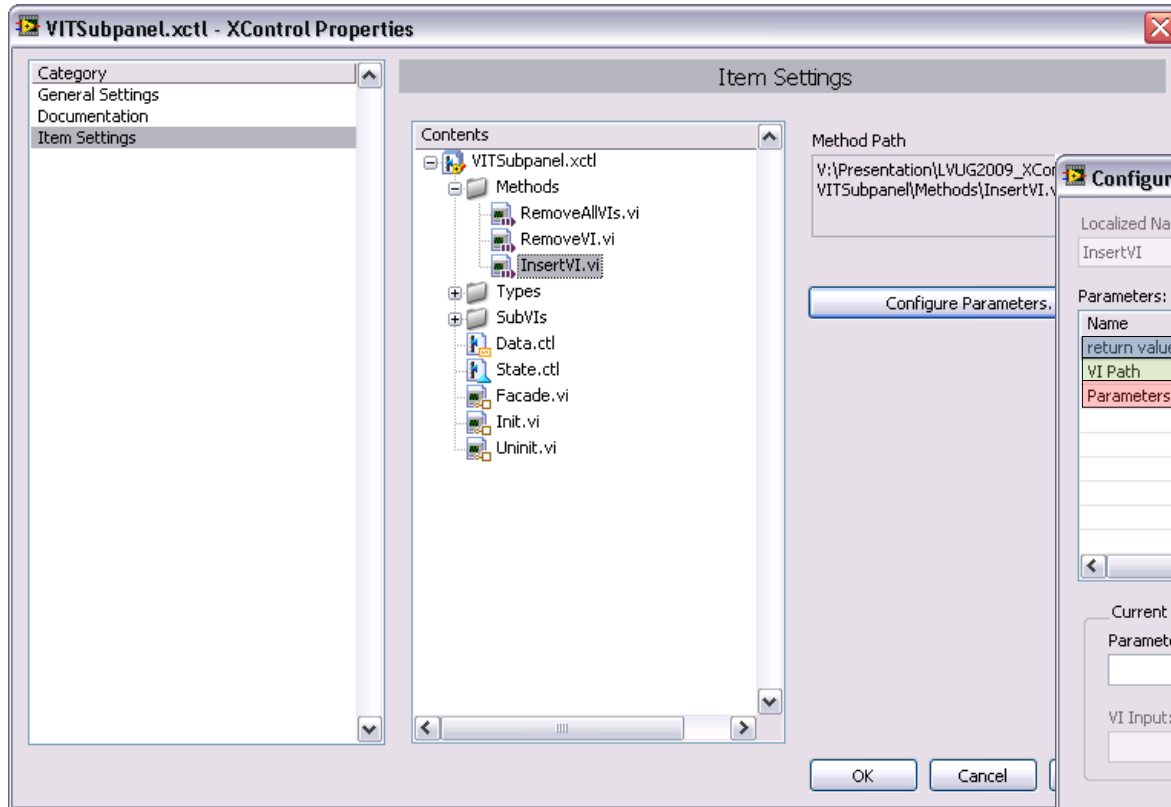
# Properties and Methods

- Configure properties



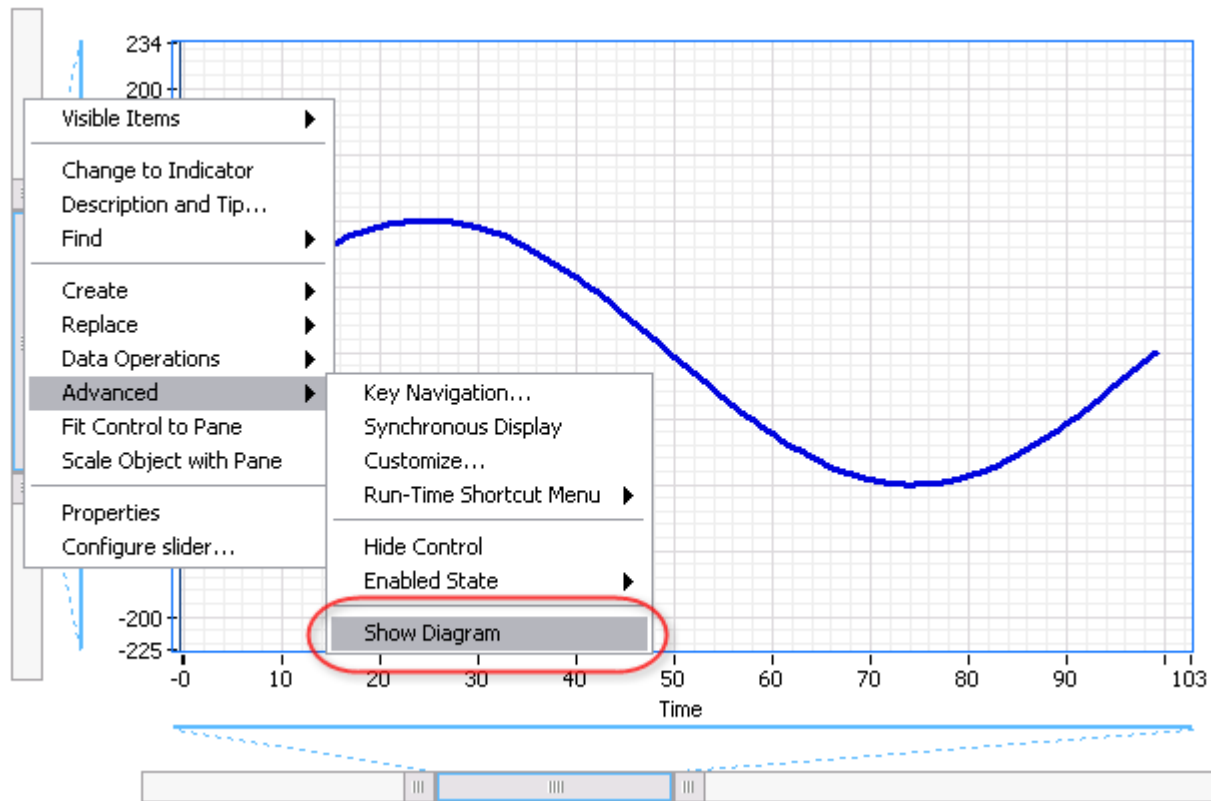
# Properties and Methods

- Configure methods
- Link method names with VI inputs and outputs



# Debugging

- Open diagram of specific XControl instance



- Use standard debugging tools and techniques

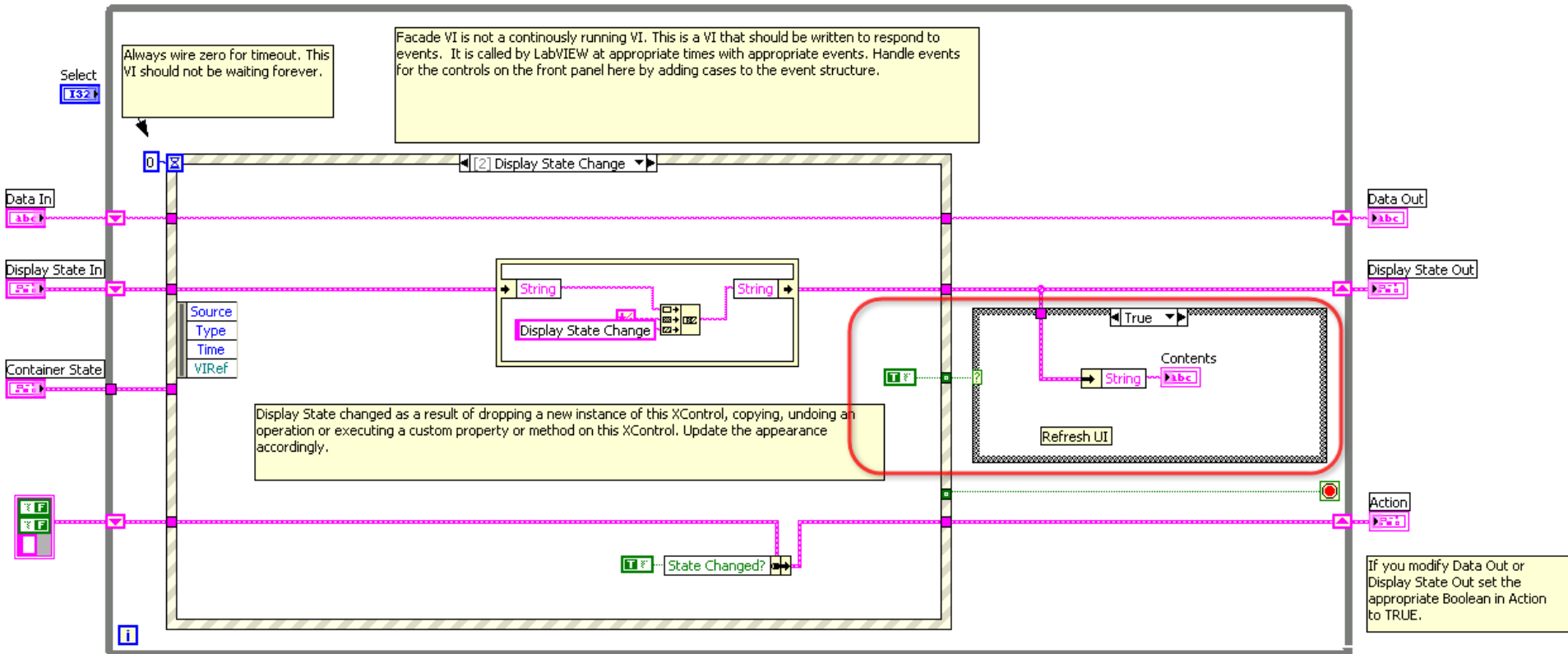


## Programming Techniques – Refresh UI

- UI update code is often needed in multiple events (e.g. Display State Change, Data Change, Resize or other FP events)
- Use Sub-VI with UI update code that is called whenever the UI needs updating
  - Potentially a lot of use of control references, which could impact performance
- Use refresh section in Facade VI



# Programming Techniques – Refresh UI





# Programming Techniques – Shortcut menus

The screenshot illustrates the implementation of a shortcut menu in LabVIEW. It consists of two main panes:

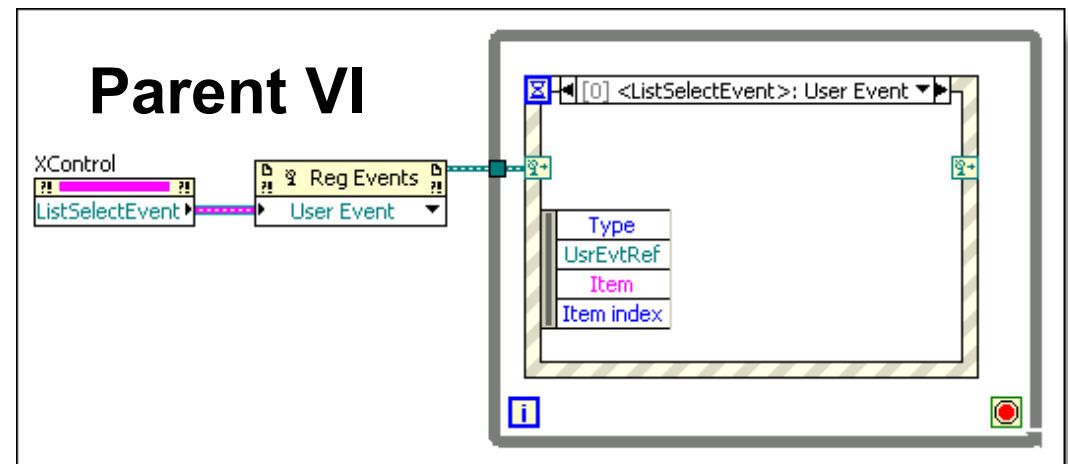
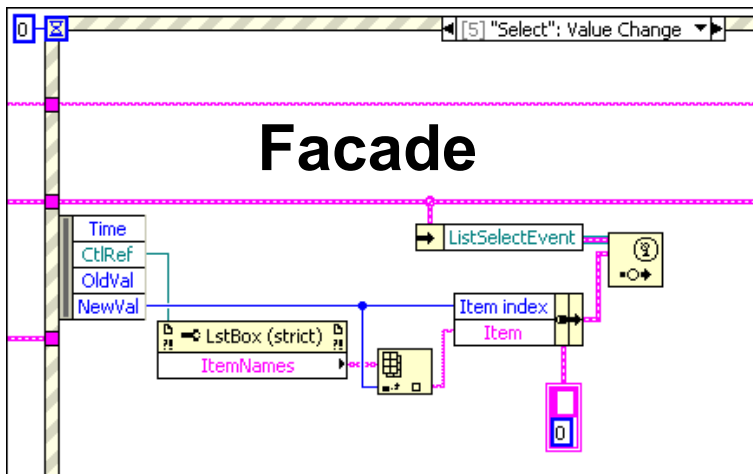
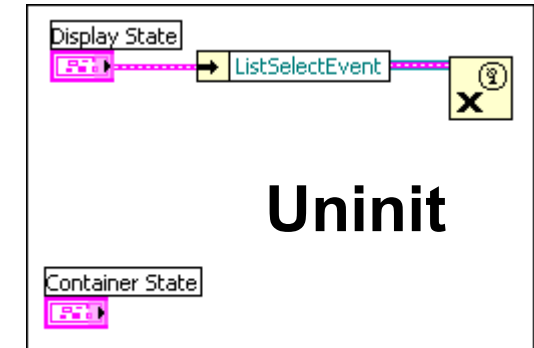
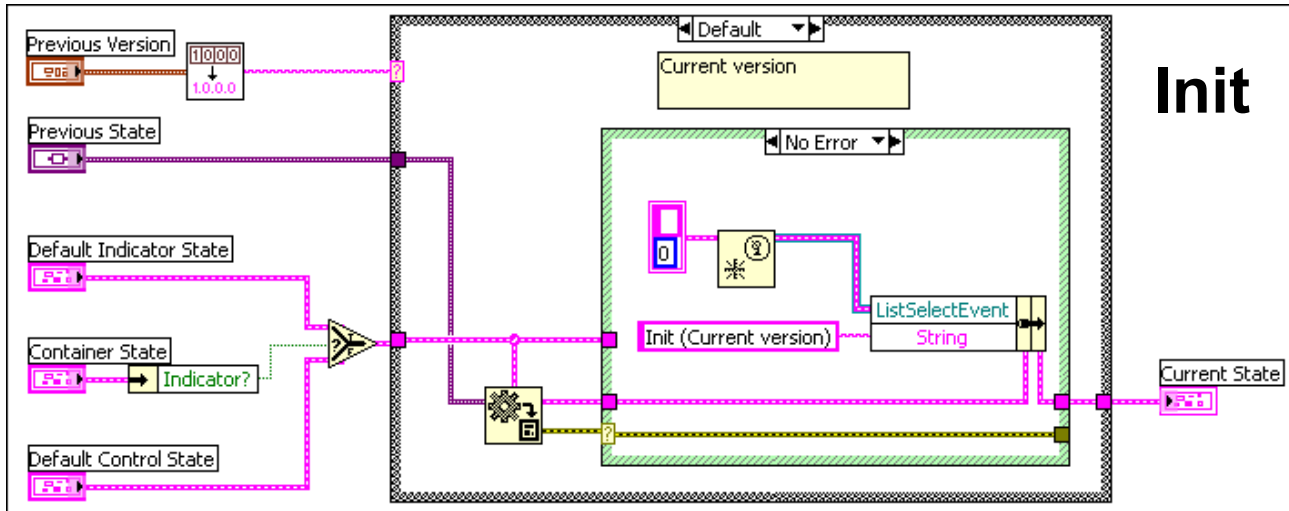
- Pane 1: "Shortcut Menu Activation?"** This pane contains a "Run Mode?" indicator. When active, it triggers a "Configure slider..." menu item. The menu is implemented using a "MENU" control and a "ConfigSlider" control.
- Pane 2: "Shortcut Menu Selection (User)"** This pane contains a "ConfigSlider" control. When the "Configure slider..." menu item is selected, it triggers a "V.Slide CONFIG DIALOG" control, which then triggers a "State Changed?" event.

A context menu is shown over the "ConfigSlider" control, with the "Configure slider..." option highlighted in red. The "Vertical ZoomSlider properties" dialog is also visible, showing settings for the slider's range and scale.

- Decide if a menu-item needs to be shown at edit-time, run-time or both
- Remember to refresh the UI if Display State is altered as a response to a menu action

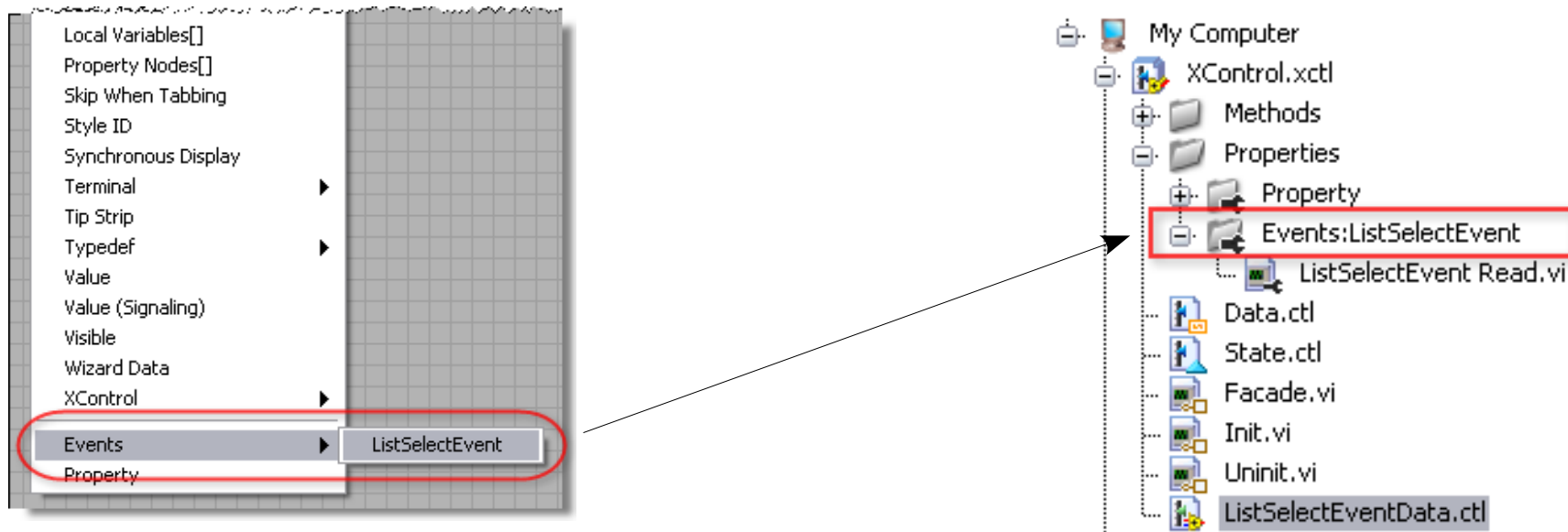
# Programming Techniques – Custom Events

- If you need the ability to notify the parent VI of actions



## Programming Techniques – Group properties and methods

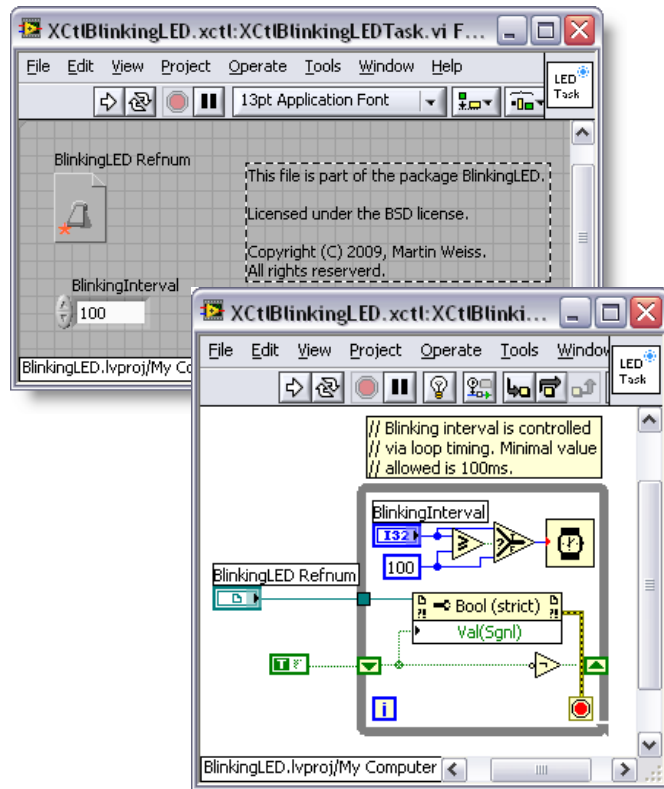
- If you have a lot of properties or methods that can be categorized together, you might want to group them in a submenu



Use colon in name to create Submenu hierarchy

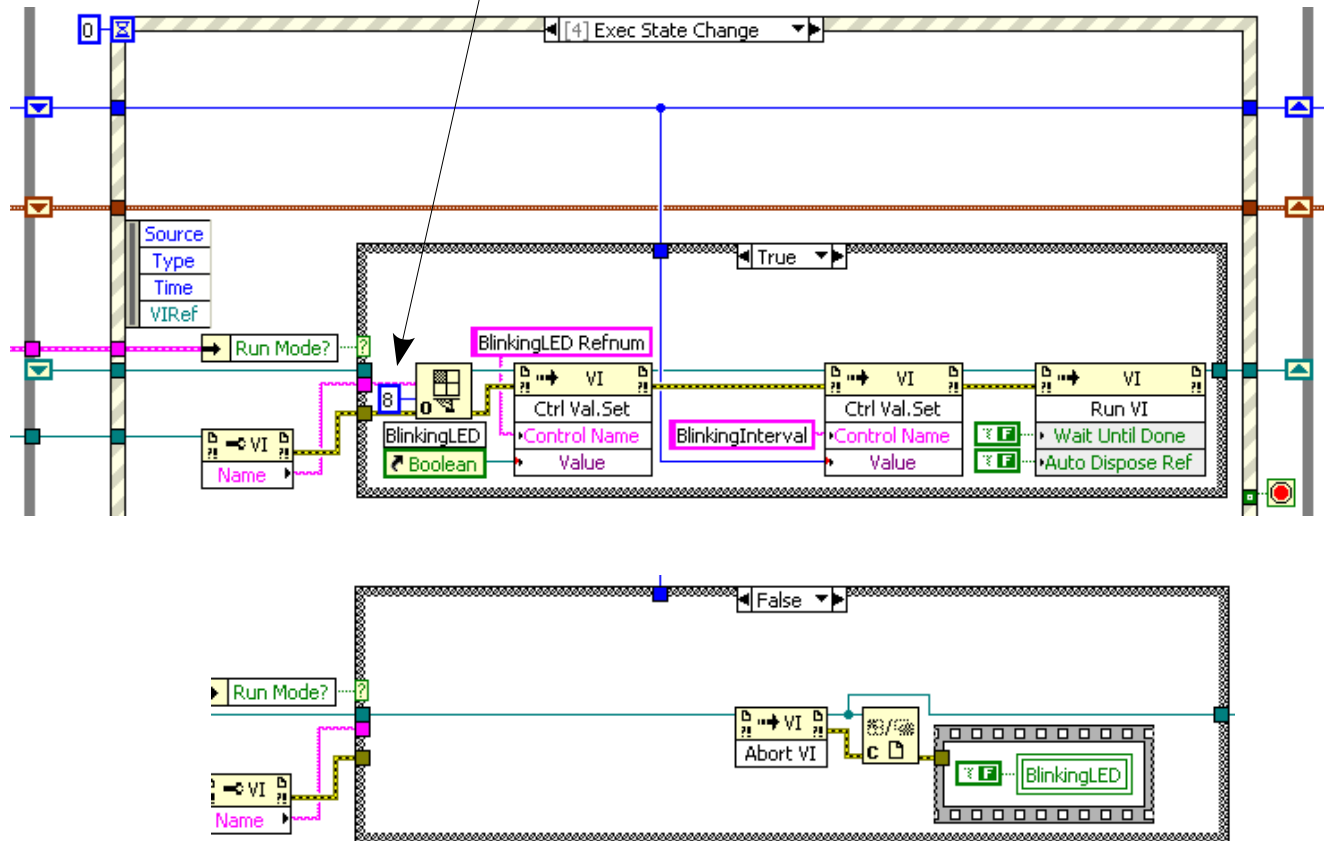
# Programming Techniques – XControl with Active process

- E.g. need animation at run-time



(Reentrant) task VI

Task VI is re-entrant, so each XControl instance gets it's own task



## Programming Techniques – XControl with Active process

- Examples with re-entrant tasks
  - Blinking LED example on LAVA CR → XControls
  - Scrolling LED XControl on NI DevZone
- Example with shared animation engine
  - AnimatorXControl (in LabVIEW examples if you have DSC toolkit installed)



# Resources

**LAVA Code Repository → User Interface → XControls**

<http://lavag.org/index.php?app=downloads&showcat=16>

**Tip on LAVA CR: XControl Inheritance tool for easy 'inheriting' of properties and methods of controls on your facade FP.**

**Scrolling LED XControl**

<http://decibel.ni.com/content/docs/DOC-1180>

**LabVIEW Help!**

**NI.COM – Search for 'XControl' and browse categories Examples, Knowledgebase and Tutorials**

**LabVIEW Advanced I: Architectures course**

