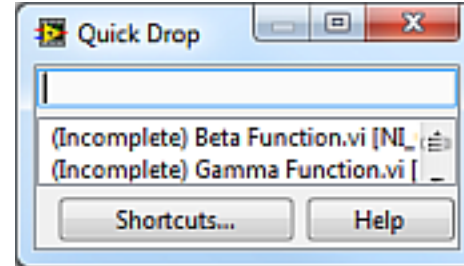


Tips and Tricks to Speed NI LabVIEW Software Development



Agenda

- Quick Drop



- Tips & Tricks for Software development



Development Tips & Tricks

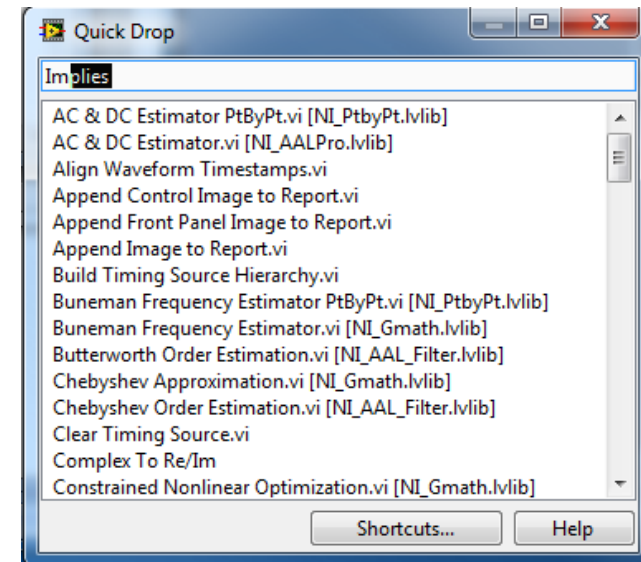
- Tips & Tricks for performance



Performance Tips & Tricks

Quick Drop

- LabVIEW 8.6 and later (but an 8.5 prototype is available)
- Drop LabVIEW objects based on name, not palette location
- Create shortcuts for commonly dropped objects
 - ...and try to make them one handed
- Functionality in LabVIEW 2009 and later:
 - Dropping project items
 - Quick Drop keyboard shortcuts
 - Wire All Terminals
 - Move Terminal Labels
 - Remove and Rewire
- Functionality in LabVIEW 2010 and later:
 - More keyboard shortcuts
 - Insert
 - Replace
 - VI Server Rename



More on Quick Drop

- Window position preserved between uses and LabVIEW sessions
 - Window is resizable in LabVIEW 2010
- Select *Tools » Options » Controls/Functions Palettes » Loading » Load palettes during launch* to make Quick Drop instantly usable
- In LabVIEW 2009 and later, add the INI token **QuickDropFastSearch=True** for faster keyboard responsiveness
- In LabVIEW 2010 and later, add the INI token **QuickDropTransparency=[num]** to assign a transparency value to the Quick Drop window

Hungry for more Quick Drop? Join the “Quick Drop Enthusiasts” group on ni.com/community

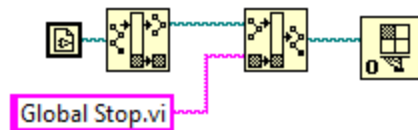
Tips & Tricks: File Path



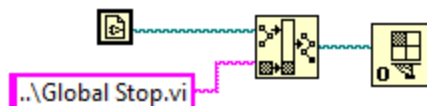
Development Tips & Tricks

LabVIEW automatically searches in current directory

Good



Better



Best

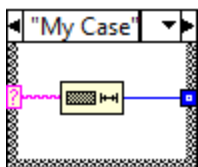


Tips & Tricks



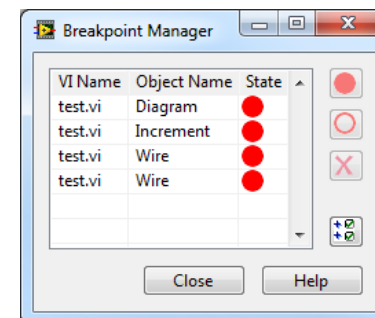
Development Tips & Tricks

Case Structure Keyboard Shortcuts (5.0)

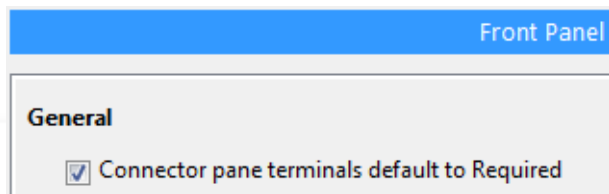


Shift-Enter – Add Case
Ctrl-Shift-Enter – Duplicate Case

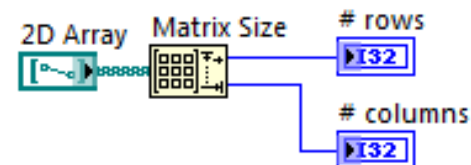
Breakpoint Manager (8.6)



Required Conpane Terminals (8.5)



Matrix Size (2009)

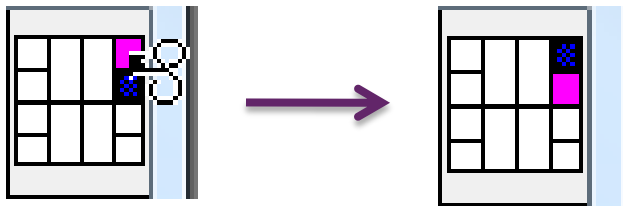


Tips & Tricks



Development Tips & Tricks

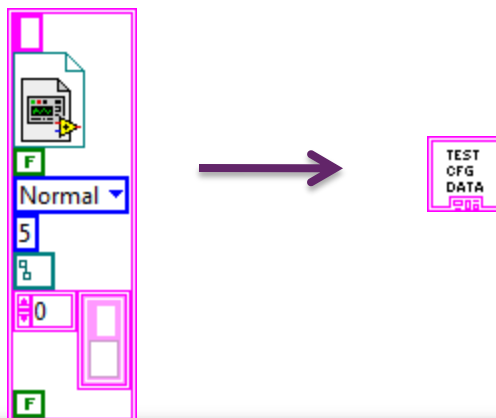
Fast Connector Pane Switching



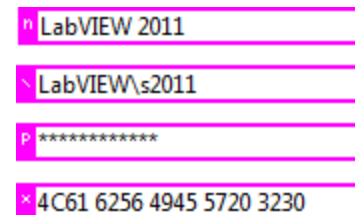
Native (Growable) Merge Errors Function



View Cluster Constant as Icon



Show Display Format for Strings

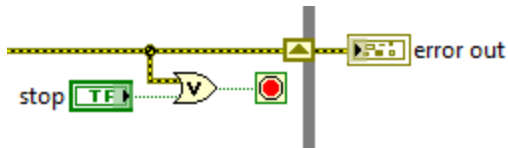


Tips & Tricks



Development Tips & Tricks

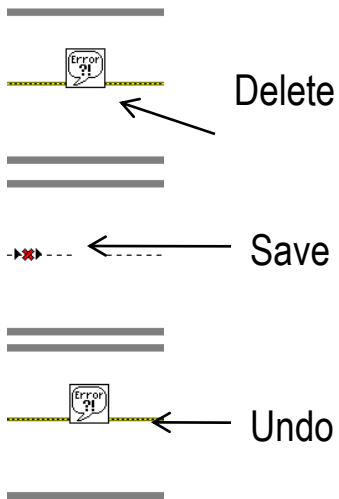
Error cluster connects with boolean inputs



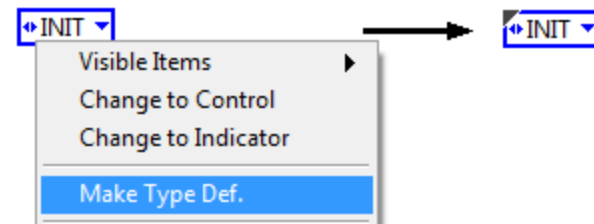
Connector Pane visible at Front Panel



Undo after the save possible (LabVIEW 2011)



Right click to create Type Def's (LabVIEW 2011)



Tips & Tricks



Better benchmarking clock (higher resolution)

High Resolution Relative Seconds.vi



..\LabVIEW 2011\vi.lib\Utility\High Resolution Relative Seconds.vi

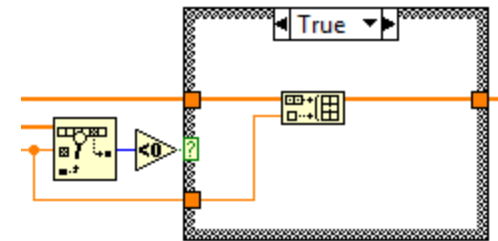
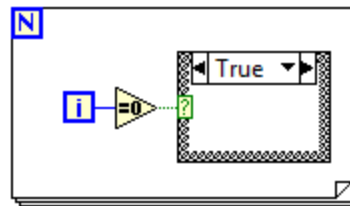
- **Changes to “Create subVI” functionality**
 - 4x2x2x4 connector pane (or another default pattern that you specify)
 - Error terminals in lower corners (and named properly)
 - Refnum/class terminals in upper corners (and named properly)
 - Clean front panel
- **Plugin Architecture through LabVIEW Scripting**
 - If you like the way we create the subVI, but you want to do something extra, you can write a plugin VI that will perform further modifications on the subVI
 - If you don’t like the way we create the subVI, you can completely replace our scripting code with your own

Tips & Tricks

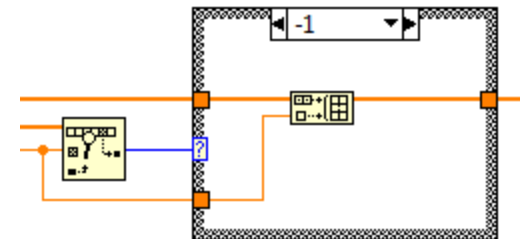
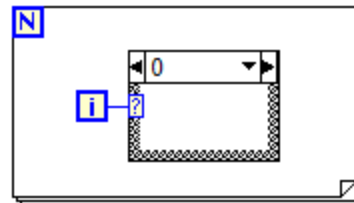


Development Tips & Tricks

Use Case Structure selector wisely!



Same result. Cleaner Block Diagram ->

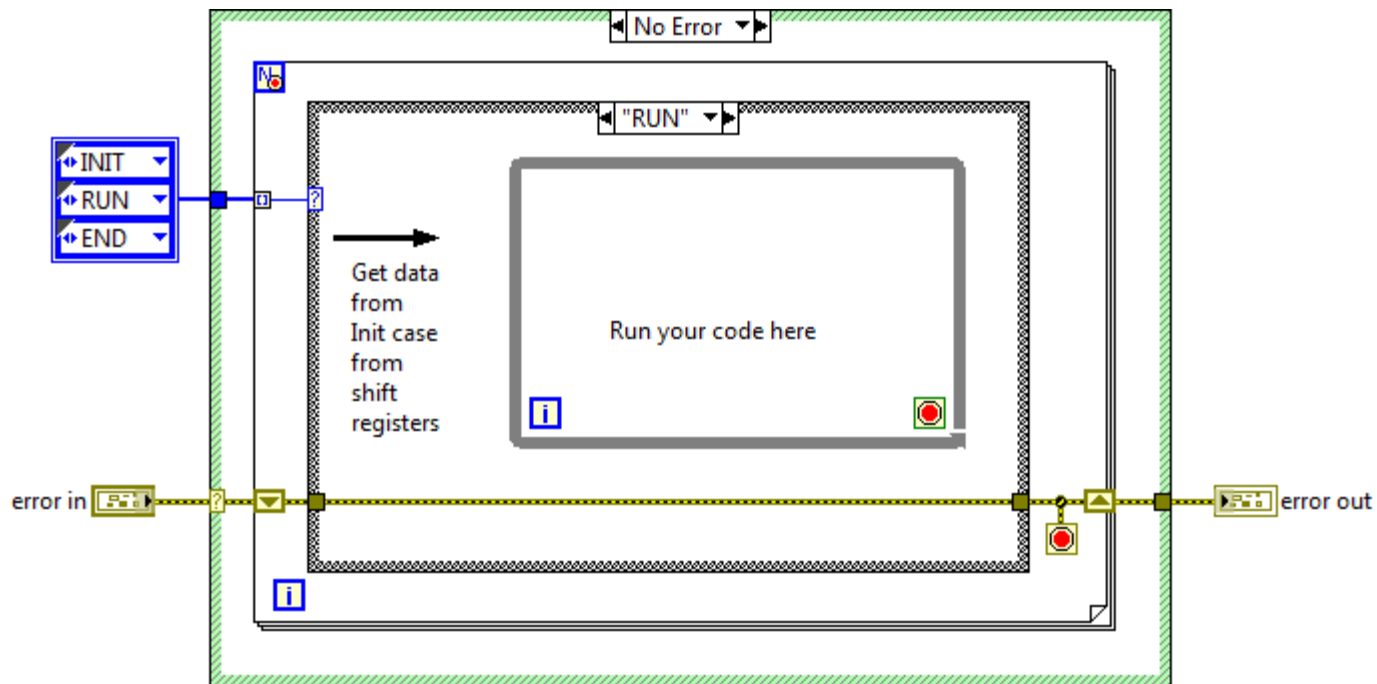


Tips & Tricks

Use templates: Example



Development Tips & Tricks

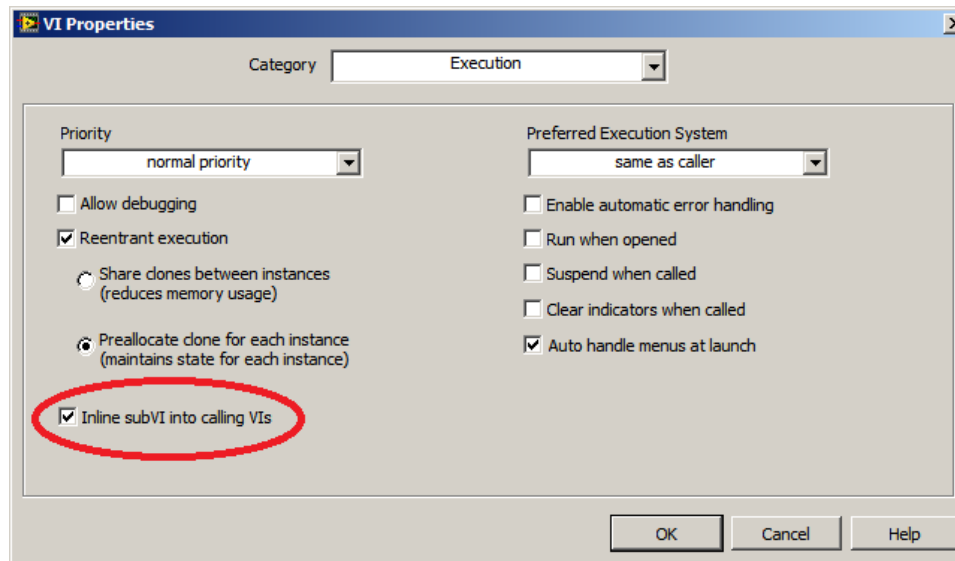


Tips & Tricks



Performance Tips & Tricks

Inline subVI into calling VI



Introduced in LabVIEW 2010, **subVI inlining** eliminates the overhead of calling subVIs by telling the compiler to act as if the subVI code resides directly on the owning diagram.

Tips & Tricks



Performance Tips & Tricks

Inline subVI into calling VI

Inlining Caveats:

- The inlined VI must be reentrant, meaning it cannot hold state information
- You cannot debug inlined VIs
- Inlining may **decrease** performance on large VIs
- Inlined VIs cannot contain recursive calls
- Inlined VIs cannot contain Property Nodes or Invoke Nodes

Tips & Tricks

Algorithm Selection

There are multiple ways to write this VI...can you figure out the *fastest* solution?

Jumble Solver

I have a list of Jumble words, and an open-source dictionary. I want to write a VI that will solve the Jumbles for me.

Jumble Example:

VABWIEL → LABVIEW



Tips & Tricks

Darren's Nuggets

<http://decibel.ni.com/content/docs/DOC-4002>

(or just search ni.com for “Darren's Nuggets”)

Thank you for attending!



Link to Demos will be sent you later ;-)