Performance Tips & Tricks

<Presenter Name Here>

NATIONAL INSTRUMENTS

We have two objectives today

1.) Tips & Tricks to make You more productive

2.) Tips & Tricks to optimize Your code performance

NATIONAL INSTRUMENTS

Tips & Tricks to make you more productive

Topics:

- Special Quick Drop shortcuts
- Plugin Architecture
- Import your tools to Functions Palette

NATIONAL INSTRUMENTS

Special Quick Drop Shortcuts

- Help you to get known functions faster
- There are some special shortcuts
  - **CTRL + T** -> Moves the control/indicator labels to the sides
  - **CTRL + I** -> Insert selected object to the wires
  - **CTRL + R** -> Removes object and fix broken wires
  - **CTRL + P** -> Replace object
  - **CTRL + B** -> Changes the VI Server type for Property/Invoke nodes
  - **CTRL + D** -> Creates controls/indicators for all unwired inputs and outputs
- You can create your own Quick Drop Shortcuts
  - We will give you one new shortcut:

  CTRL + N -> Creates a new VI + “Error/No Error” case structure + Icon

DEMO

NATIONAL INSTRUMENTS

Plugin Architecture

- Software Platform that you can reuse with multiple projects
- Standardizes your code
- Good performance
- Flexibility
- Scalability

NATIONAL INSTRUMENTS

Plugin Architecture

The main Structure

Core VIs

Use these VIs with all projects. This is the Main Sequencer code and tools that are for general use.

+

Plugin VIs

Use these VIs only with current project.

NATIONAL INSTRUMENTS

Plugin Architecture

The Folder Structure

Keep the current project related files separated from the general SW. Do not mix them!

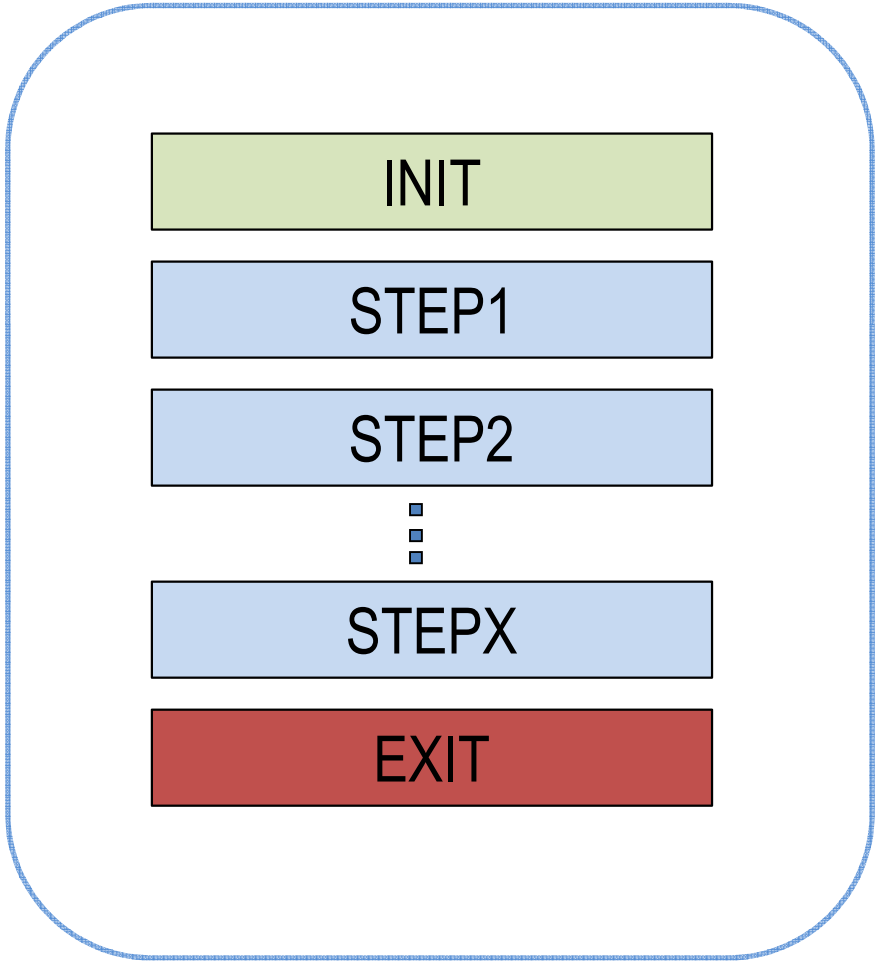
Use Source Code Control. Create different repository for General Tools and for current project.

DEMO

NATIONAL INSTRUMENTS

Plugin Architecture

The Sequencer

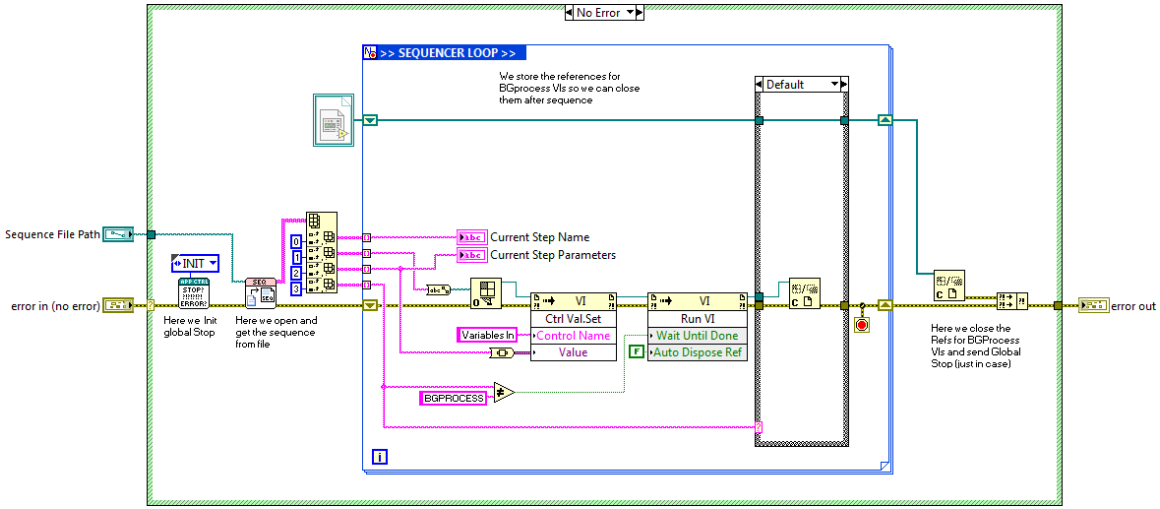


DEMO

NATIONAL INSTRUMENTS

Plugin Architecture

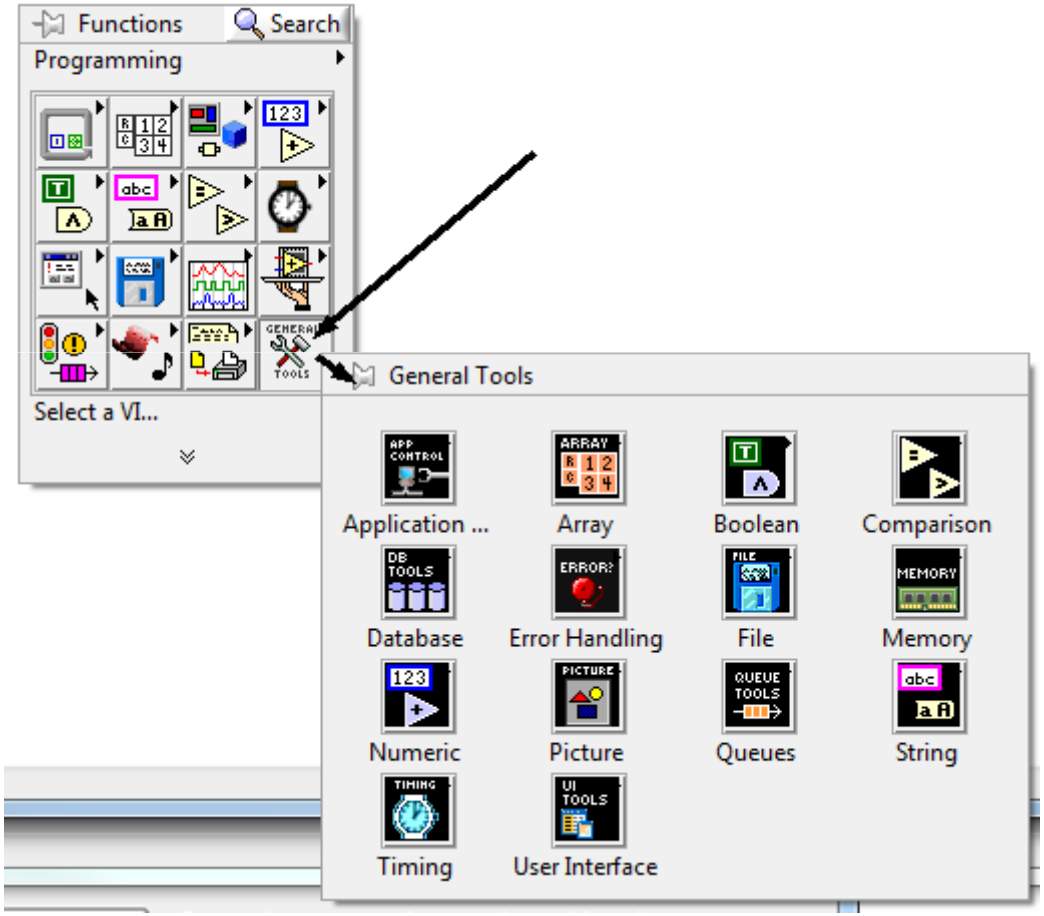
Simple Plugin Based Sequencer Demo



DEMO

NATIONAL INSTRUMENTS

Import your own tools to Functions Palette



DEMO

NATIONAL INSTRUMENTS

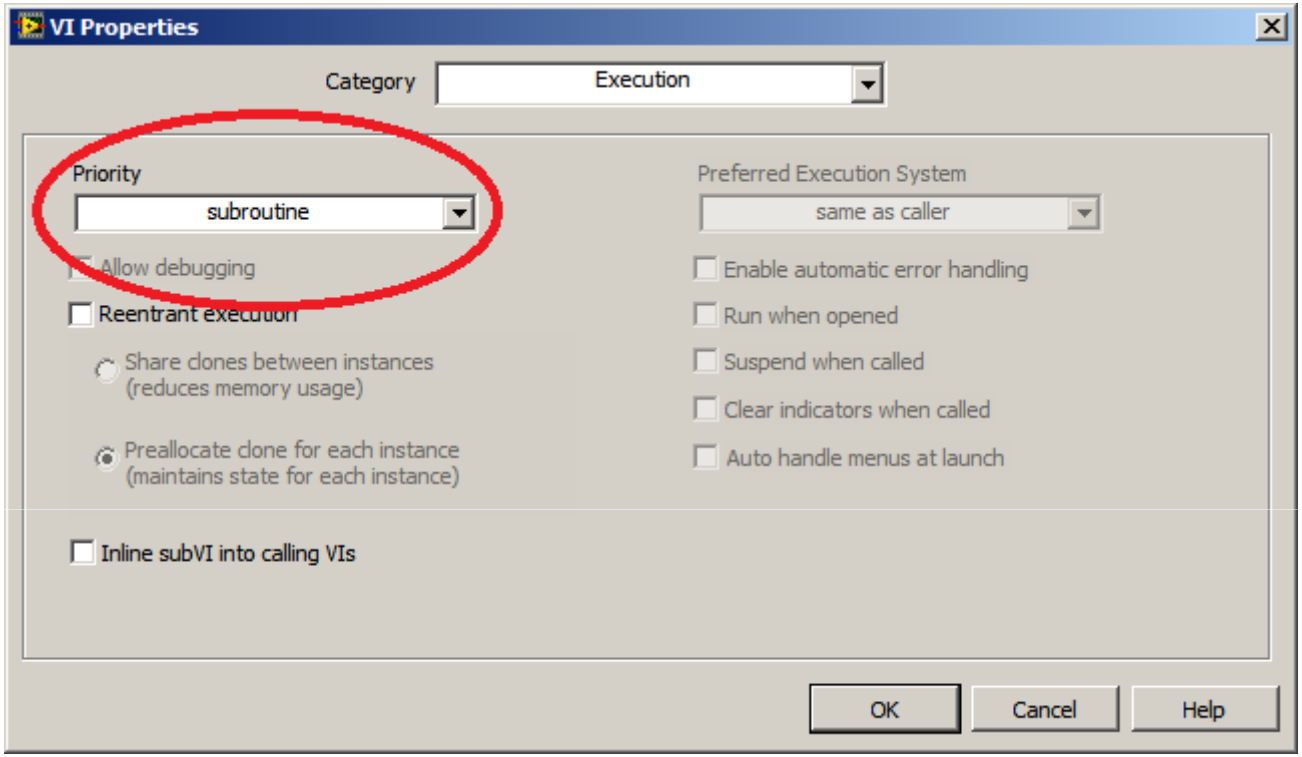
Performance Tips & Tricks

Topics:

- Subroutine Priority
- Easy Dynamic Calls
- Array Tips & Tricks

NATIONAL INSTRUMENTS

Subroutine Priority



The **subroutine** priority setting on a VI causes that VI to take control of the thread in which it is executing. This allows it to run as efficiently as possible.

DEMO

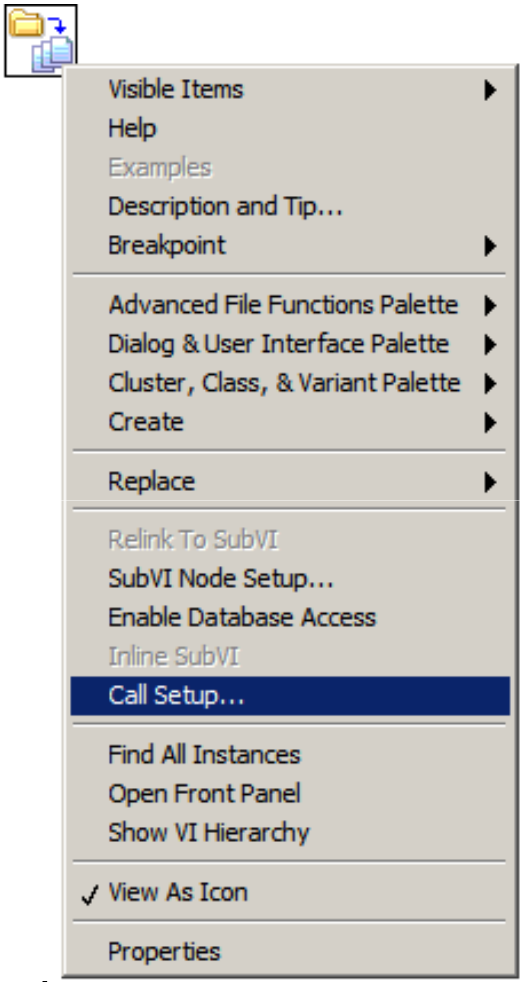
NATIONAL INSTRUMENTS

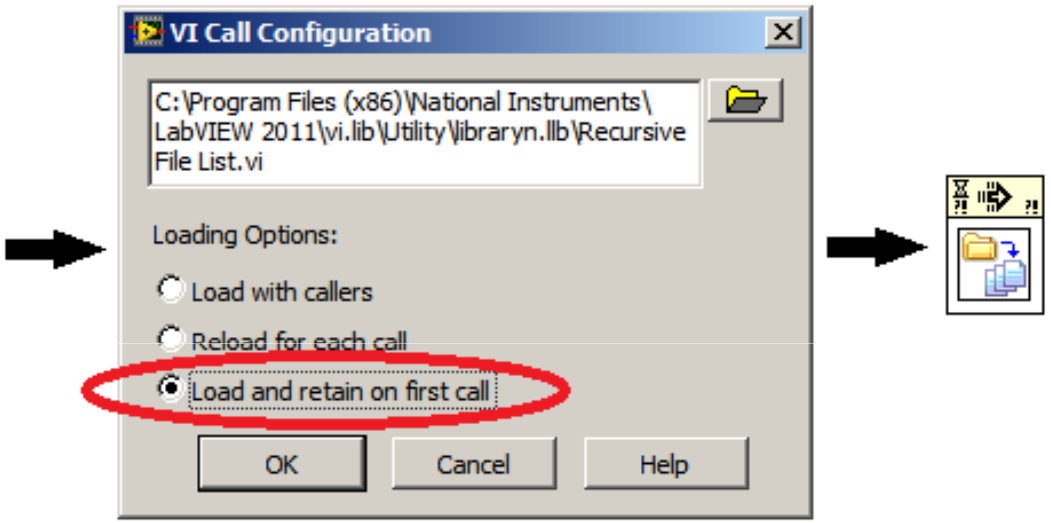
Subroutine Priority

Subroutine Caveats:

1. A subroutine VI can only call other subroutine VIs
2. A subroutine VI cannot call any blocking functions (Wait, One Button Dialog, VISA calls, etc.)
3. Front Panel controls and indicators are not updated during execution
4. No other VI in the calling VI's thread can run while a subroutine VI is running

Easy Dynamic Calls





The “Call Setup” feature (introduced in LabVIEW 8.0) makes it very easy to change a static subVI call into a dynamic call to improve load time performance.

DEMO

NATIONAL INSTRUMENTS

Easy Dynamic Calls

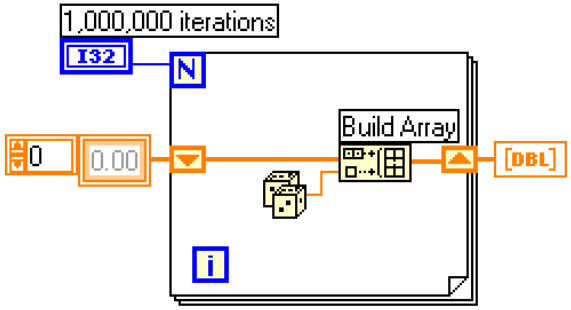
Dynamic Call Caveats

- If the calling VI is in edit mode, all dynamic VIs will be in memory
- The “VI Call Configuration” dialog displays an absolute path, but the calling VI stores a relative path
- “Reload for each call” should only be used if you need to release the memory allocated for each subVI call

NATIONAL INSTRUMENTS

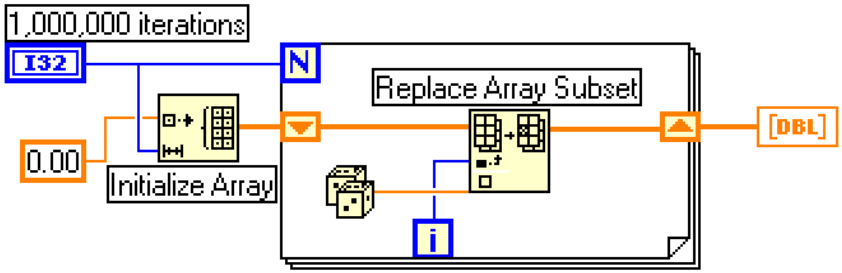
Array Tips & Tricks

Build an Array in **18.7** Seconds



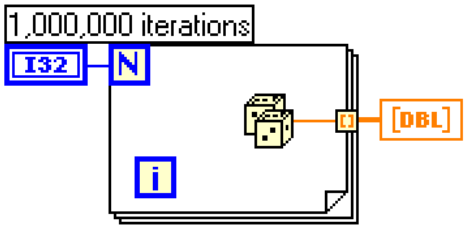
Very slow because each loop iteration involves a memory reallocation

Build an Array in **0.42** Seconds



Much faster because there is a single memory allocation

Build an Array in **0.40** Seconds



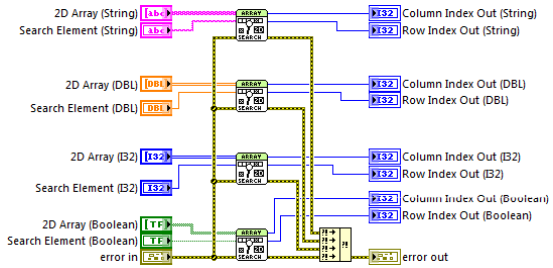
Fastest method is also the cleanest

NATIONAL INSTRUMENTS

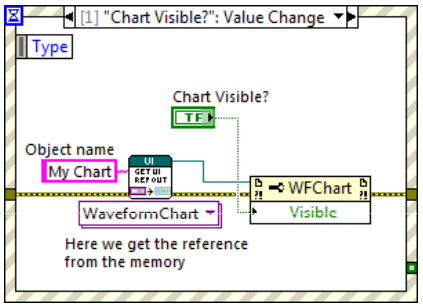
Something Extra ;-)

Some Useful Tools:

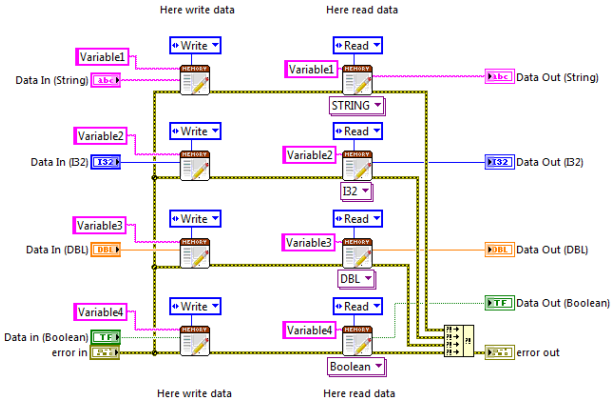
Search 2D Array



Function to store UI Obect Refences



Data storage function



DEMO

NATIONAL INSTRUMENTS

Thank you for participating
LabVIEW Programmer Days 2012

NATIONAL INSTRUMENTS