NATIONAL INSTRUMENTS™

LabVIEW User Group Meeting

Long Island Chapter

2 September 2010

Agenda

- Introductions
- Upcoming Events
- What’s New in LV 2010

Robert Berger

- BS Electrical Engineering from Texas A&M
- Joined NI’s Engineering Leadership Program in 2001
- Supported and trained customers for ~4 years
- Migrated to Long Island in April 2007
- Covers Long Island and NYC
- Available for demos, onsite seminars, technical consultation, specification assistance, loaner equipment…

Upcoming Events

- Local seminars
  - On-sites at any time
  - LV for Embedded Applications Hands-On September 8
  - User Group, 1st Thursday of Dec.
- Training
  - LabVIEW Core 3, Sept 13, Melville, NY
  - LabVIEW Softwae Engineering, Sept 16, Melville, NY
  - On-line Weekly

- NI-Week
  - Conference, August 2011
- NI Certification
  - Available at local testing sites

What's New in LabVIEW 2010

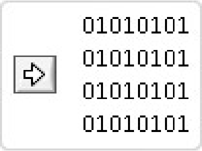
Robert Berger

Sr. District Sales Manager, National Instruments

What's New in LabVIEW 2010?

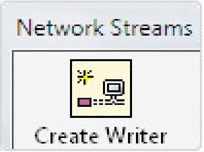
- Improved Performance
- LabVIEW Environment Enhancements
- Large Application Development
- Improved Data Transfer and Distributed HW Configuration
- What's New in the LabVIEW Modules
- Building LabVIEW Add-ons

New Features in LabVIEW 2010



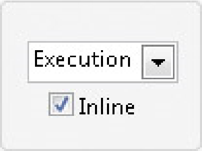
Compiler Optimizations

Run your code up to 20 percent faster with new back-end compiler technologies and custom code optimizations



Stream Data over the Network

Stream data continuously between LabVIEW applications using the new Network Streams API

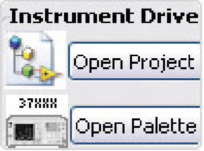


SubVI Inlining

Improve your application's performance by removing the overhead associated with subVI calls using this new execution option

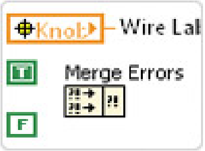
Web-Based Hardware Configuration

Set up and maintain your remote hardware with a router-like configuration experience



Find and Install Instrument Drivers

Shorten the time to first measurement by easily finding and installing more than 10,000 certified instrument drivers

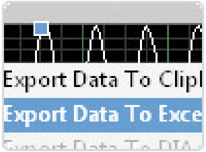


Feedback-Driven Improvements

Facilitate development with 14 new features based on user feedback

Save VIs without Compiled Code

Simplify source code management by separating the compiled objects from the actual LabVIEW source code

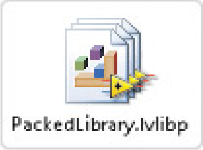


Export Graph Data to Excel

Easily export your data to Microsoft Excel with a simple right-click option

Import External FPGA IP

Ease FPGA development by importing Xilinx Core Generator DSP IP into LabVIEW



Packed Project Libraries

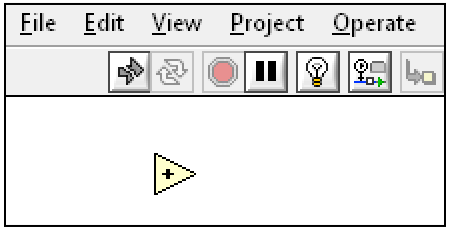
Simplify code deployment and distribution by packaging your source code into a single file with packed project libraries

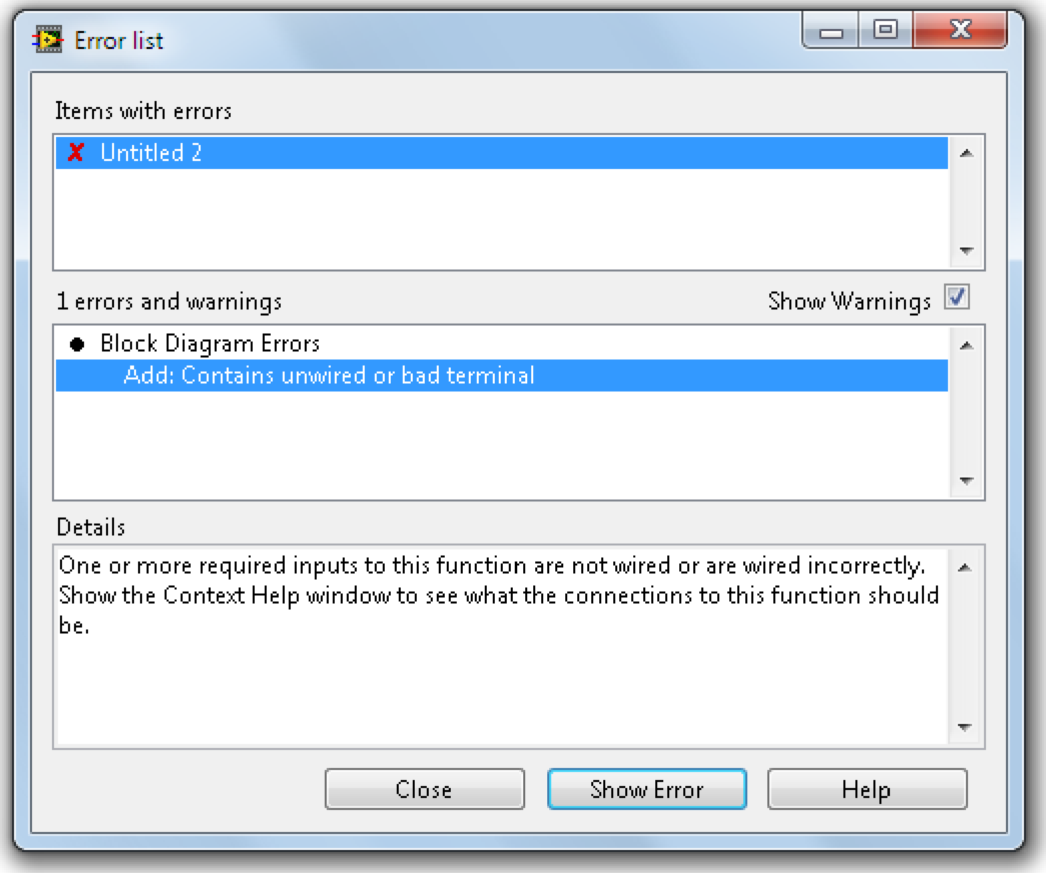
IMPROVED PERFORMANCE

LabVIEW Compiler

Abstracts the complexities of programming

- Memory management
- Thread allocation
- Language syntax





Items with errors

☒ Untitled 2

1 errors and warnings

Show Warnings ☑

● Block Diagram Errors

Add: Contains unwired or bad terminal

Details

One or more required inputs to this function are not wired or are wired incorrectly. Show the Context Help window to see what the connections to this function should be.

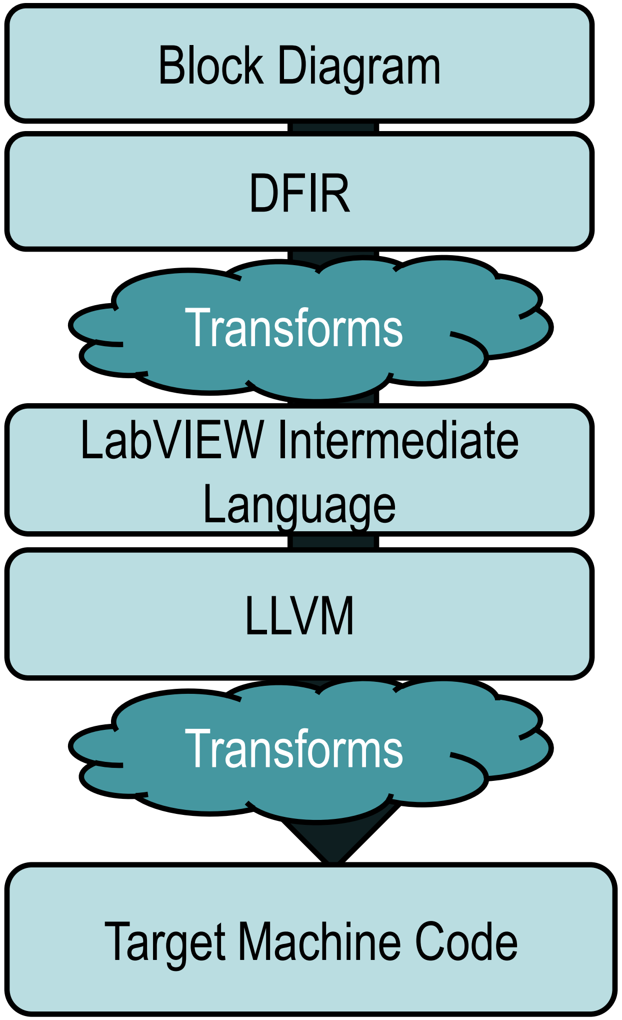
Optimizing the LabVIEW Compiler

DataFlow Intermediate Representation (DFIR)

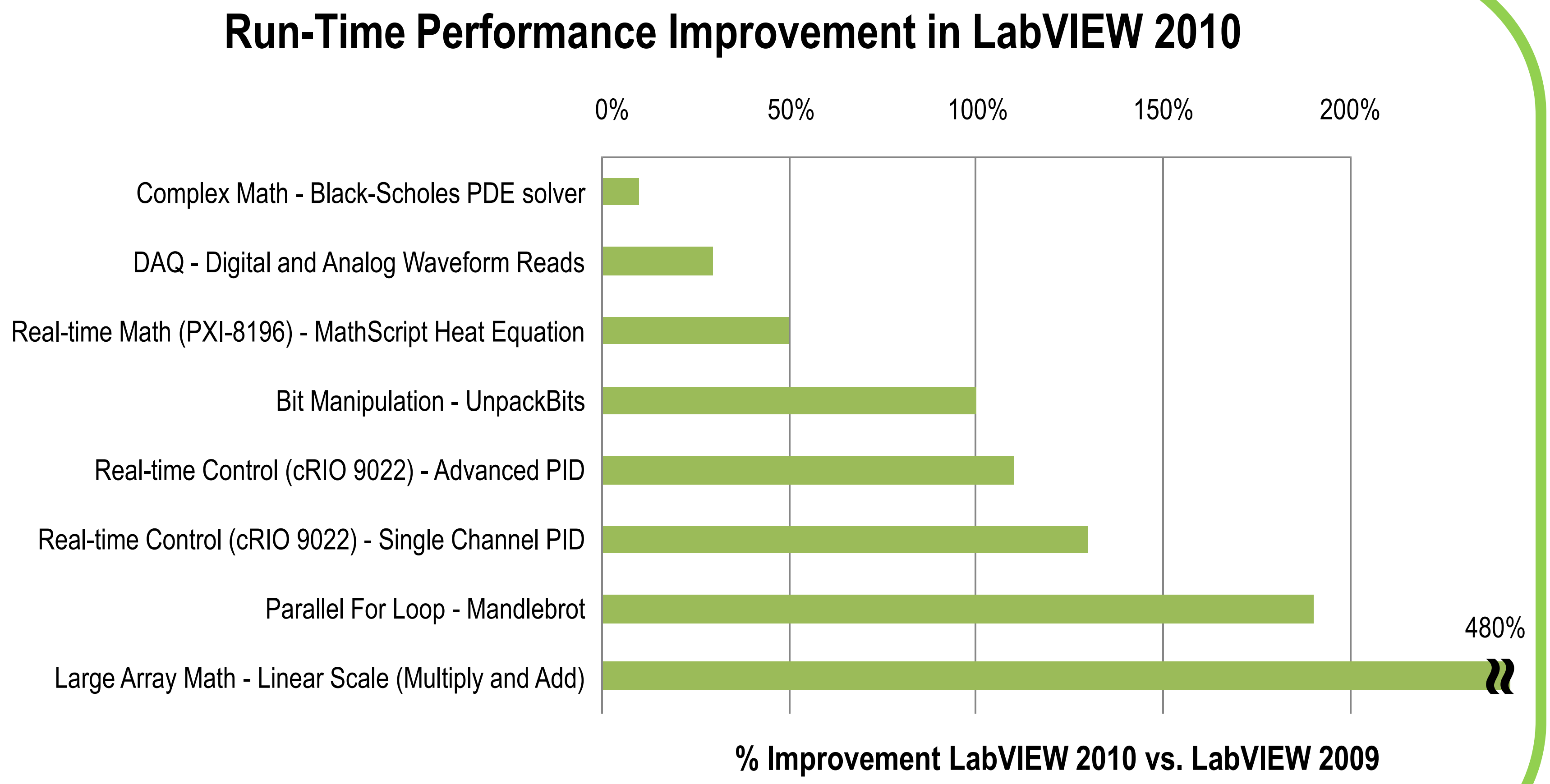
- High-level representation
- Preserves dataflow, parallelism, and execution semantics

Low-Level Virtual Machine (LLVM)

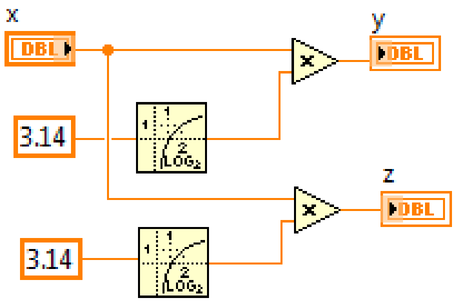
- Low-level representation
- Sequential
- Knowledge of target machine characteristics, instruction sets, alignment, etc.

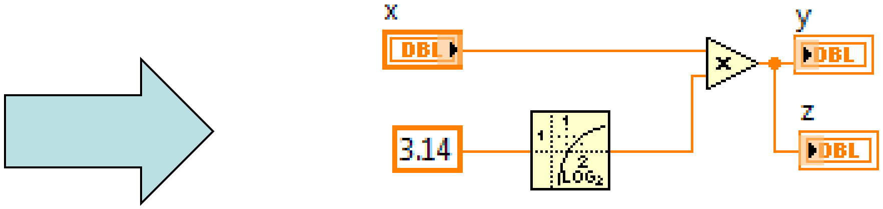


LabVIEW 2010 Performance Metrics

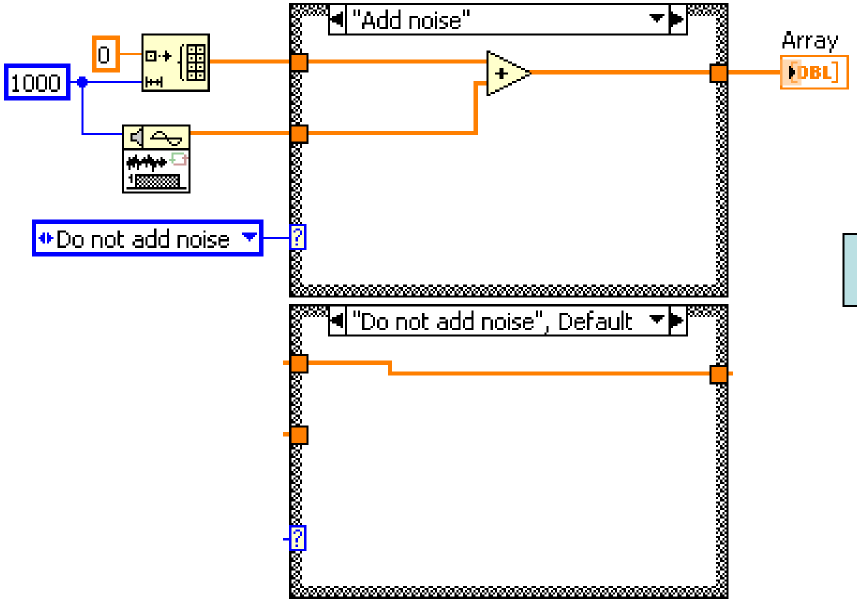


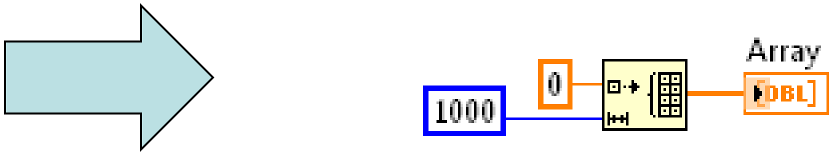
LabVIEW Compiler Decompositions





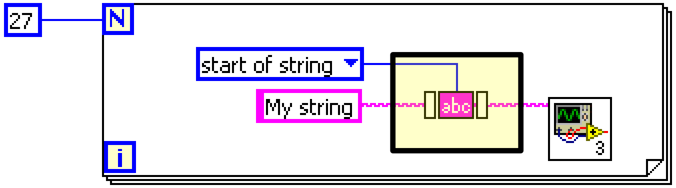
Common Subexpression Elimination

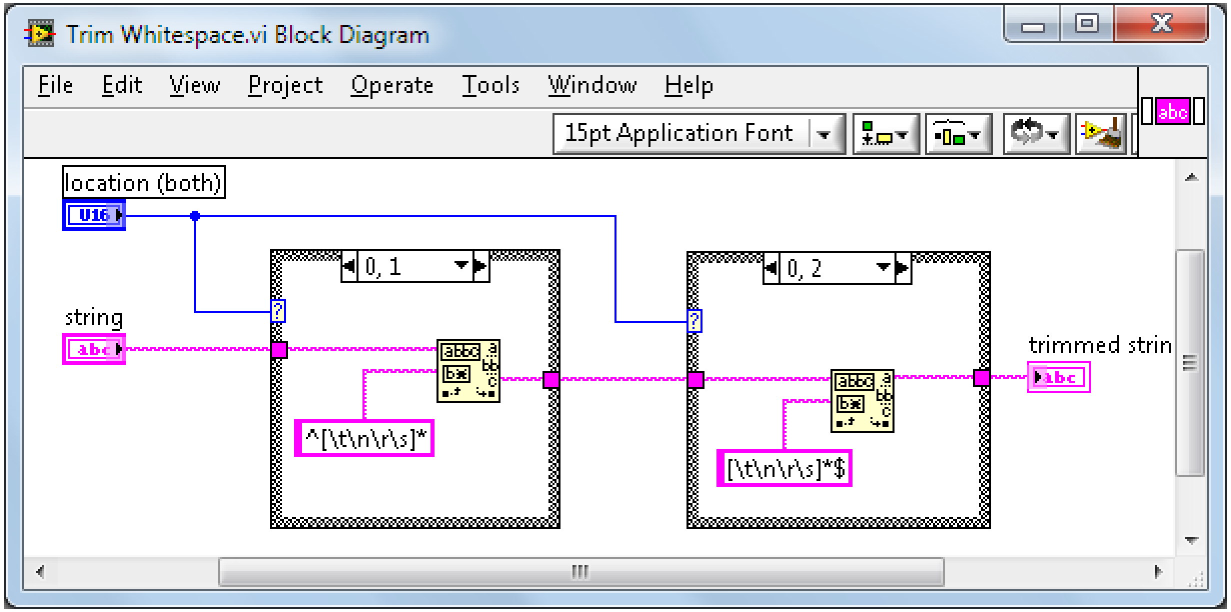




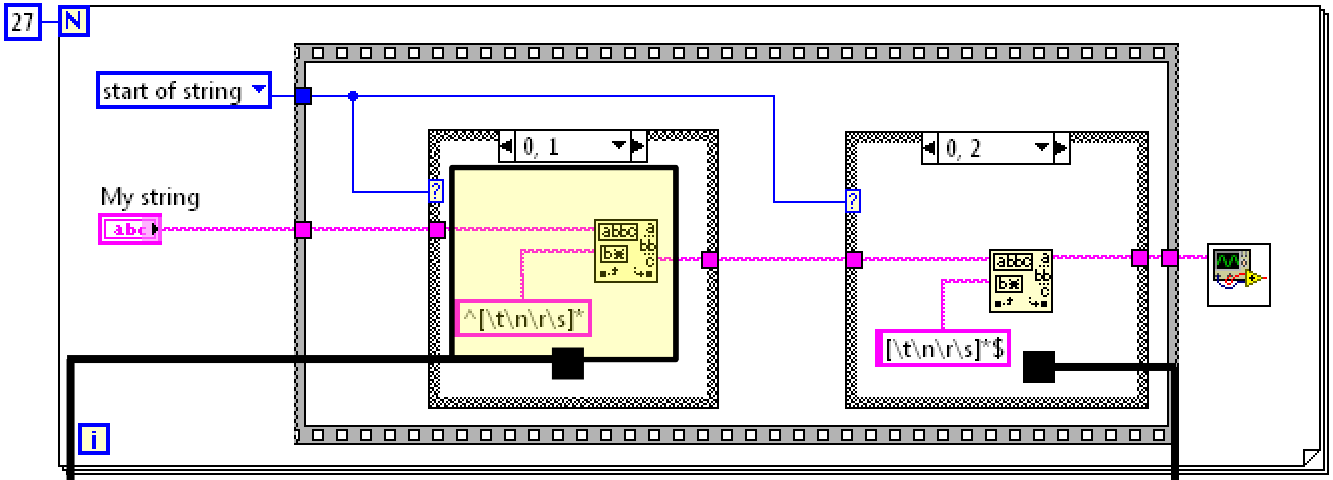
Unreachable Code Elimination

LabVIEW Compiler Optimizes Your Code





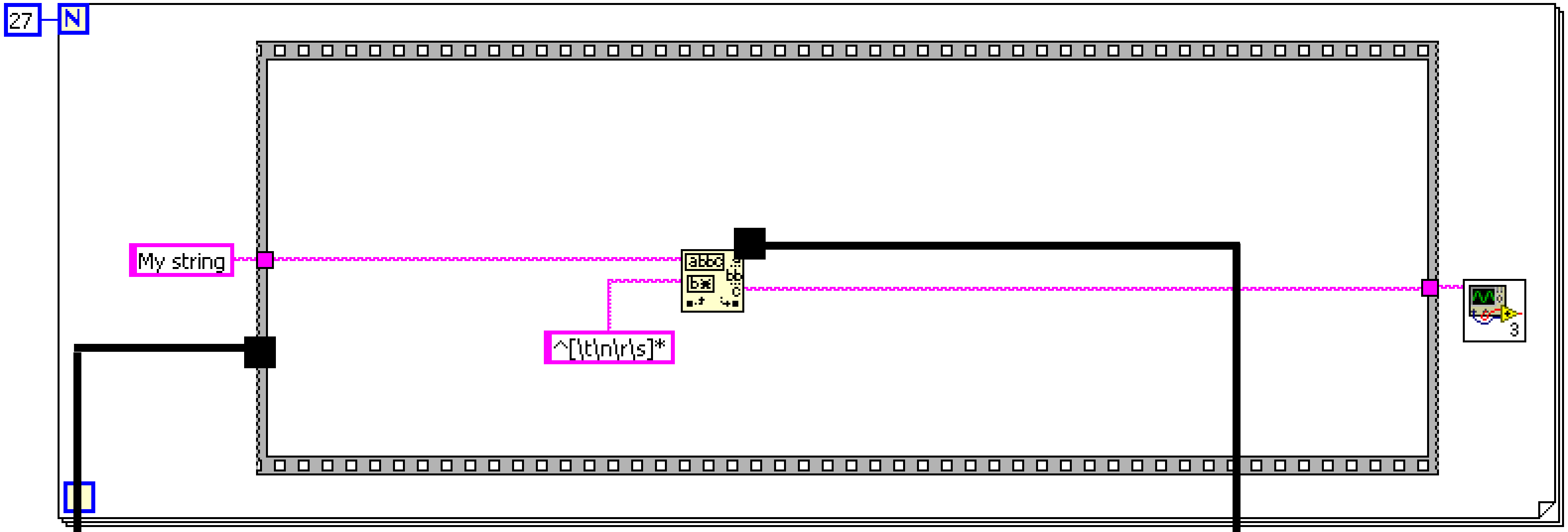
LabVIEW Compiler Optimizes Your Code



Only this portion of the code will execute

Because the input is constant, the compiler can determine which code will execute, and remove the unnecessary code

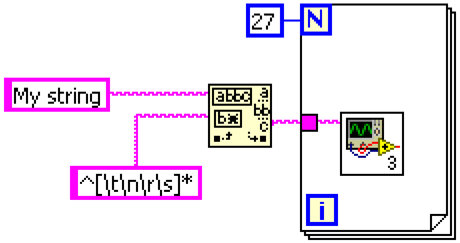
LabVIEW Compiler Optimizes Your Code



Sequence Structure is now unnecessary

The Match Pattern primitive will not change from iteration to iteration

LabVIEW Compiler Optimizes Your Code

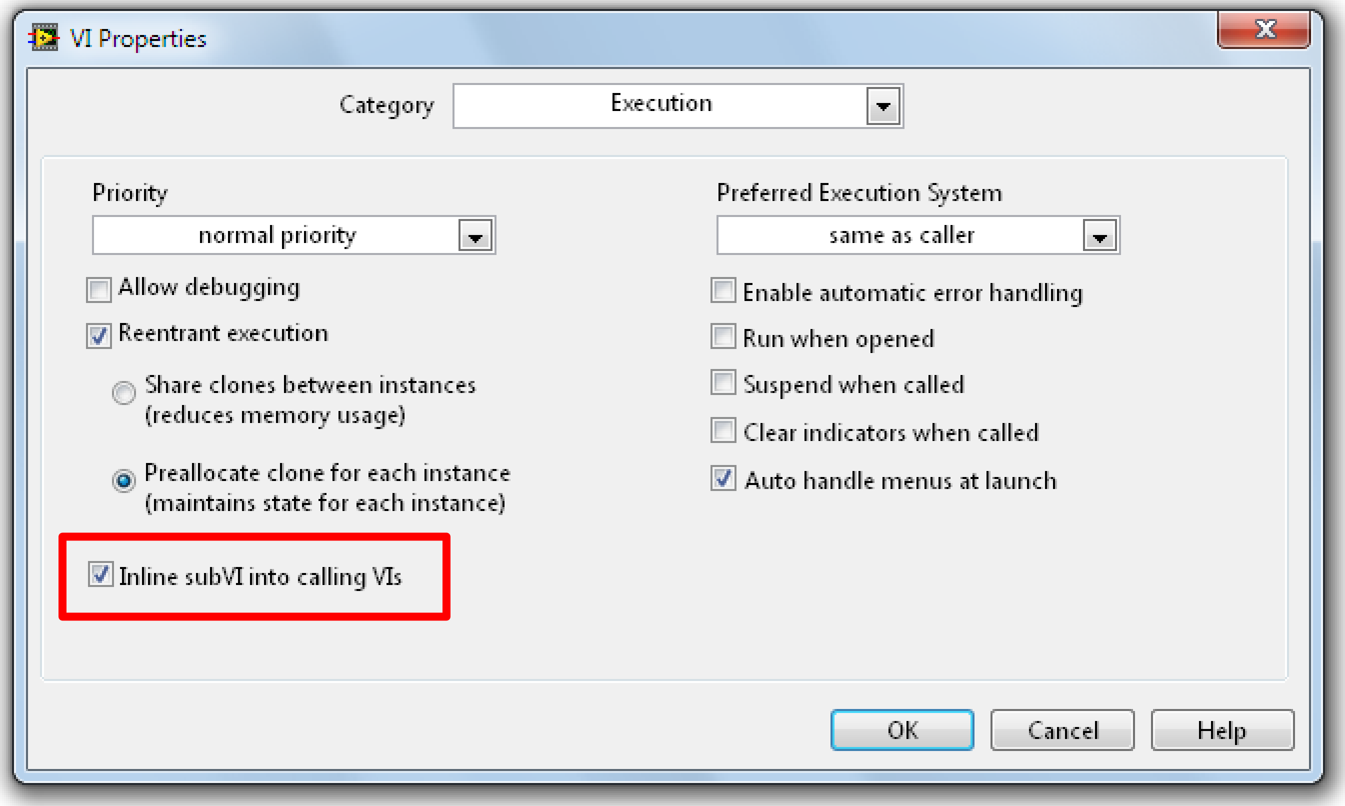


Decompositions Used

- SubVI Inlining
- Unreachable Code Elimination
- Dead Code Elimination
- Loop Invariant Code Motion

SubVI Inlining

Maintain Code Modularity With Minimum Overhead



VI Properties

Category: Execution

Priority: normal priority

□ Allow debugging

☑ Reentrant execution

○ Share clones between instances (reduces memory usage)

● Preallocate clone for each instance (maintains state for each instance)

☑ Inline subVI into calling VIs

Preferred Execution System: same as caller

□ Enable automatic error handling

□ Run when opened

□ Suspend when called

□ Clear indicators when called

☑ Auto handle menus at launch

- Removes all subVI call overhead
- Automatically updates callers when callee’s code changes

LabVIEW 2010 Performance Metrics

What Won’t Get Faster?

Hardware Calls

DLL Calls / Optimized C Code

User Interface Interaction

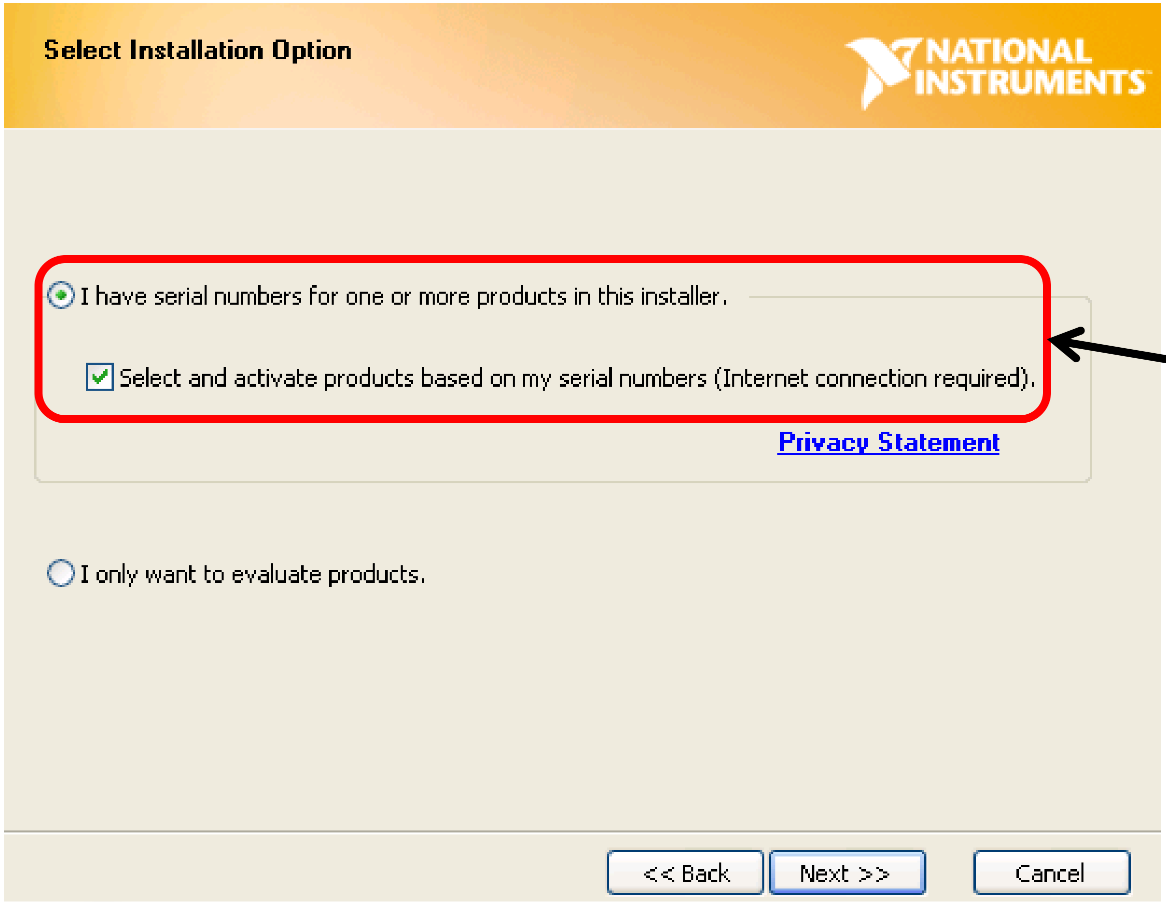
Increased Compile Time

Compile Time	5x
Mass Compile Time	35%
Application Build Time	35%

ENVIRONMENT ENHANCEMENTS

Smarter Installer

Select Your Software Based on Serial Number(s)



Select Installation Option

● I have serial numbers for one or more products in this installer.

☑ Select and activate products based on my serial numbers (Internet connection required).

○ I only want to evaluate products.

Enter your Serial Number(s), and the installer will determine what to install

LabVIEW Idea Exchange

288
Kudos!

New Boolean Diagram constant design!

Labels: UI & Usability   Status: In Beta

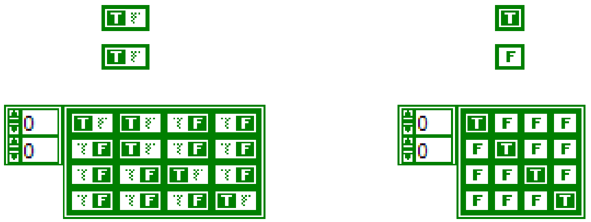
by altenbach   07-03-2009 04:37 PM

The current boolean diagram constant is potentially confusing and too elaborate.

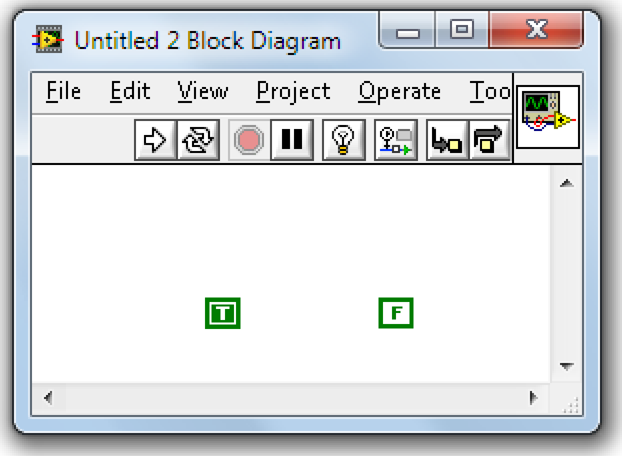
Confusing, because it almost looks like a toggle switch, so the new user might click on the **right half**, expecing an unconditional FALSE. However, there are no active areas, and an inversion of the current value occurs no matter where we click.

Too elaborate. All we need to see is the current value! Why do we need to see the "other" value greyed out??? We can guess that by simple elimination. 😉 There is too much redundant information, wasting twice as much diagram space than actually needed to display relevant information. The current design also makes e.g. 2D boolean diagram constant very confusing. Have a look at the image. Can you immediately tell that the 2D array on the left is only true on the diagonal? (I did not think so!). Now look at the suggestion on the right. Ahh... much better! 😀

Suggestion:
The boolean diagram constant should be **smaller, simpler, and cleaner**.
The image shows the current design on the left and the suggested design on the right.

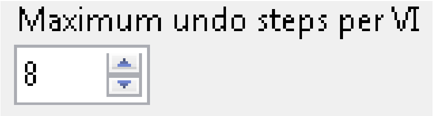
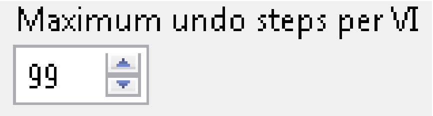
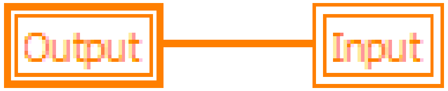
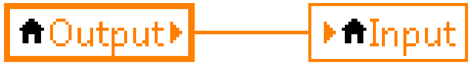
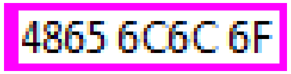
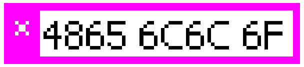
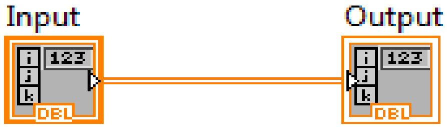
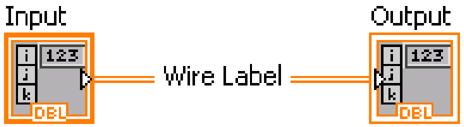
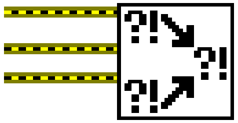
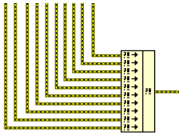
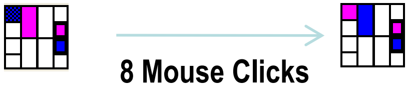
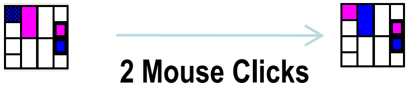


Current   Suggested



What a difference in clarity and economy!!

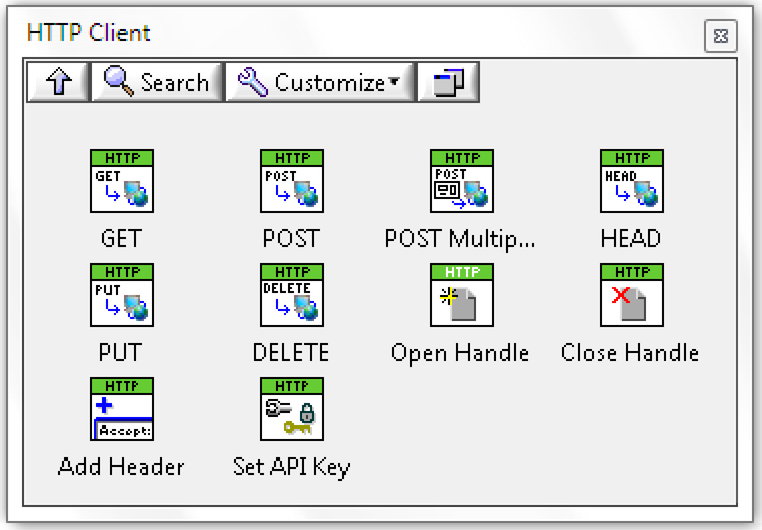
LabVIEW 2010 Idea Exchange

Feature Name	LabVIEW 2009	LabVIEW 2010	User
Default Number of Undo Steps	 Maximum undo steps per VI 8	 Maximum undo steps per VI 99	*PJM_LabVIEW*
Local Variable Redesign			*Altenbach*
String Radix			*Altenbach*
Wire Labels			*Falkpl*
Growable Merge Error Node			*Dany.*
Move Switch Items in the connector pane	 8 Mouse Clicks	 2 Mouse Clicks	*tst*

DEMO

USER REQUESTED FEATURES

HTTP(S) Nodes

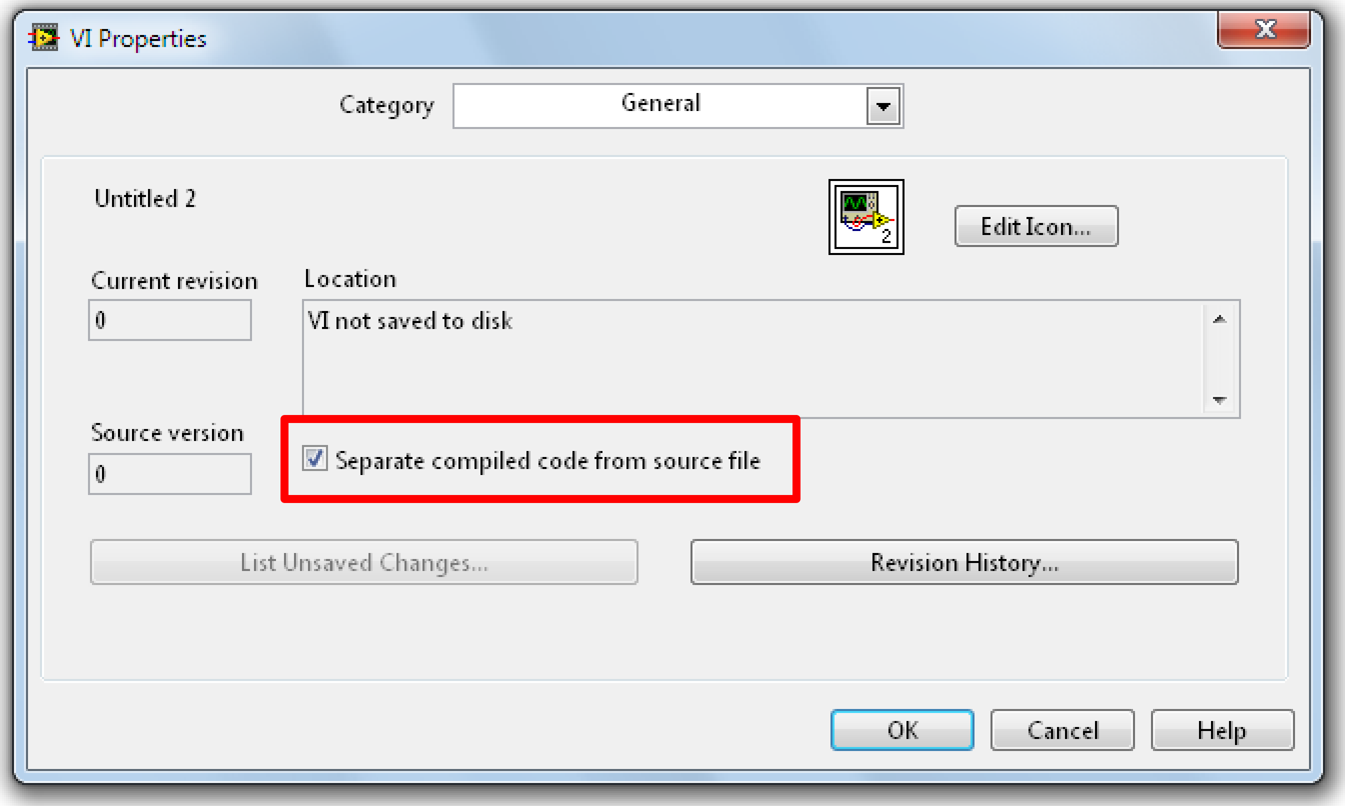


- Use the new **HTTP Client VIs** to build a LabVIEW web client
- Interact with servers, Web pages, and Web services
- Works with LabVIEW or 3rd-party Web services

LARGE APPLICATION DEVELOPMENT

Separate Compiled Code From Source File

Improved Source Code Control



VI Properties

Category: General

Untitled 2

Current revision: 0

Location: VI not saved to disk

Source version: 0

☑ Separate compiled code from source file

Eliminate the need to re-save and re-submit files to source code control unless the graphical source code has been changed by the developer

Packed Project Libraries

Distribute and Reuse LabVIEW Code Easily

- Deploy the VI hierarchy with a single file
- Shorten build times for calling VIs
- Simplify code deployment
- .lvlibp file type

Example	# Source VIs	EXE Build Time	# VIs Built Into PPL	EXE Build Time	Build Time Improvement
Agilent 34401 Acquire and Graph - SW Triggered.vi	53	6.3 s	22	5.15 s	18.2%
E-Mail Notification.vi	102	8.66 s	68	5.82 s	32.8%
Update Weather Data.vi	71	12.97 s	46	5.48 s	57.8%
Custom Example	1000	53.93 s	999	15.94 s	70.4%

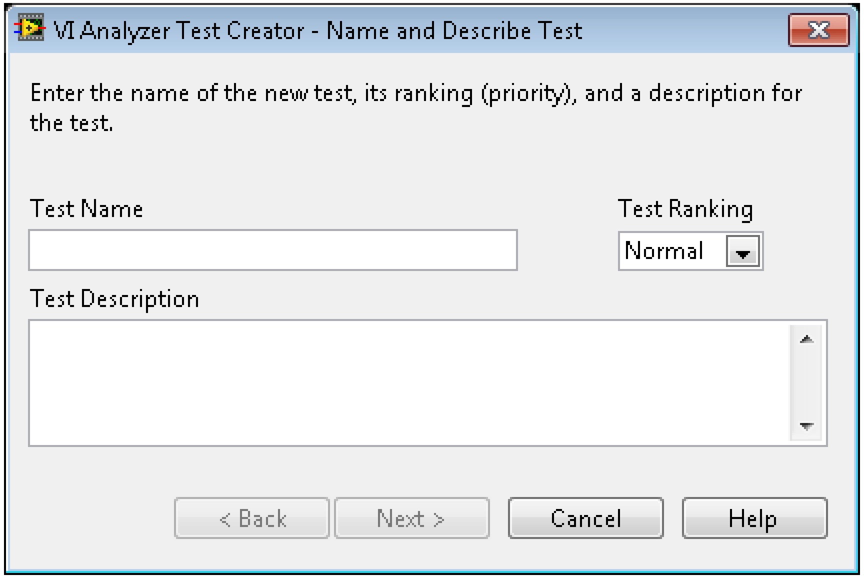
LabVIEW Add-Ons for Software Validation

Unit Test Framework Toolkit

- 30% faster test execution
- Custom definition of test vector ranges

VI Analyzer Toolkit

- Create you own tests in VI Analyzer using LabVIEW Scripting



VI Analyzer Test Creator - Name and Describe Test

Enter the name of the new test, its ranking (priority), and a description for the test.

Test Name:

Test Ranking: Normal

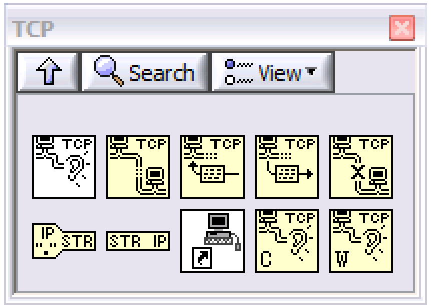
Test Description:

TARGET-TO-HOST DATA TRANSFER

Network Connectivity Options in LabVIEW

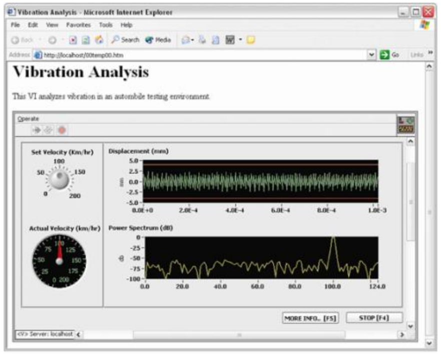
TCP/IP and UDP

Define low-level communication protocols



Remote Front Panels

Quickly embed a front panel in a browser

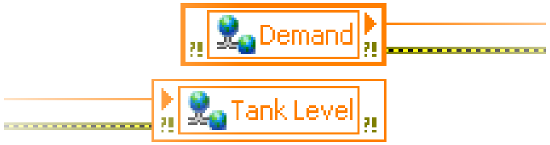

Vibration Analysis - Microsoft Internet Explorer

Vibration Analysis

This VI analyzes vibration in an autombile testing environment.

Shared Variables

Quickly develop distributed systems through drag-and-drop configuration



DEMO

TARGET-TO-HOST DATA TRANSFER

What’s New

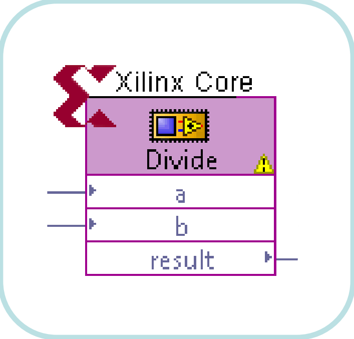
LABVIEW 2010 MODULES

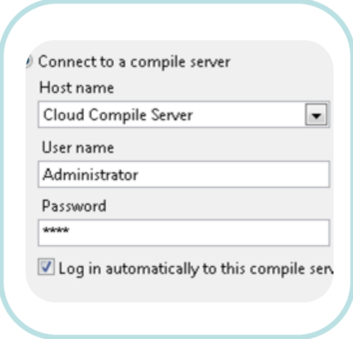
LabVIEW 2010 Real-Time Products

- **LabVIEW Real-Time Module**
  - Web-based configuration and monitoring of networked targets
  - Simpler host-to-target transfer of data using Network Streams
  - Publish variables via Web Services
  - Software IEEE 1588 as timing source for Timed Loop

- **NI-Real-Time Hypervisor 2.0**
  - Shared memory for higher data transfer rates between OSs
  - Higher customization for CPU partitioning
  - Added Linux support

LabVIEW 2010 FPGA Module



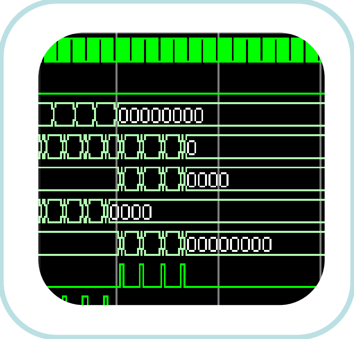

Connect to a compile server

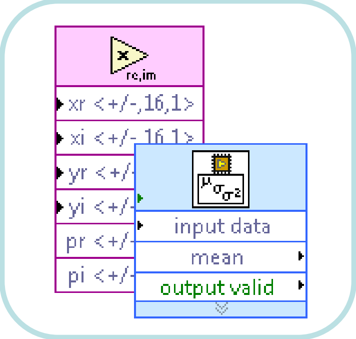
Host name: Cloud Compile Server

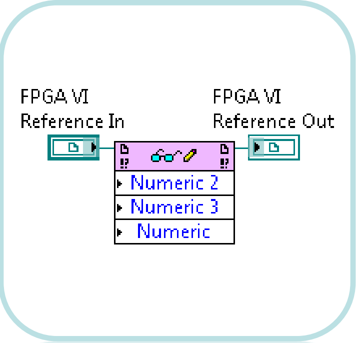
User name: Administrator

Password: ****

☑ Log in automatically to this compile serv







IP Integration Node - Directly import Xilinx .xco files or your own VHDL easily

New Compilation Flow - Earlier Compilation Estimates and Build Specifications

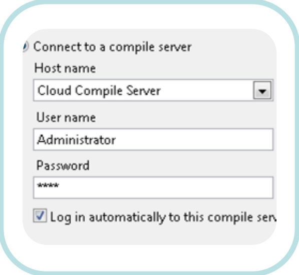
Cycle-Accurate Simulation - Use ModelSim for Cycle-Accurate Simulation

More IP Blocks - New IP for Statistics, Complex Multiplication, and More

Host Improvements – New Dynamic reference for Host VI reuse

LabVIEW 2010 FPGA Compilation

LabVIEW FPGA Compile Farm Toolkit



● Connect to a compile server

Host name

Cloud Compile Server

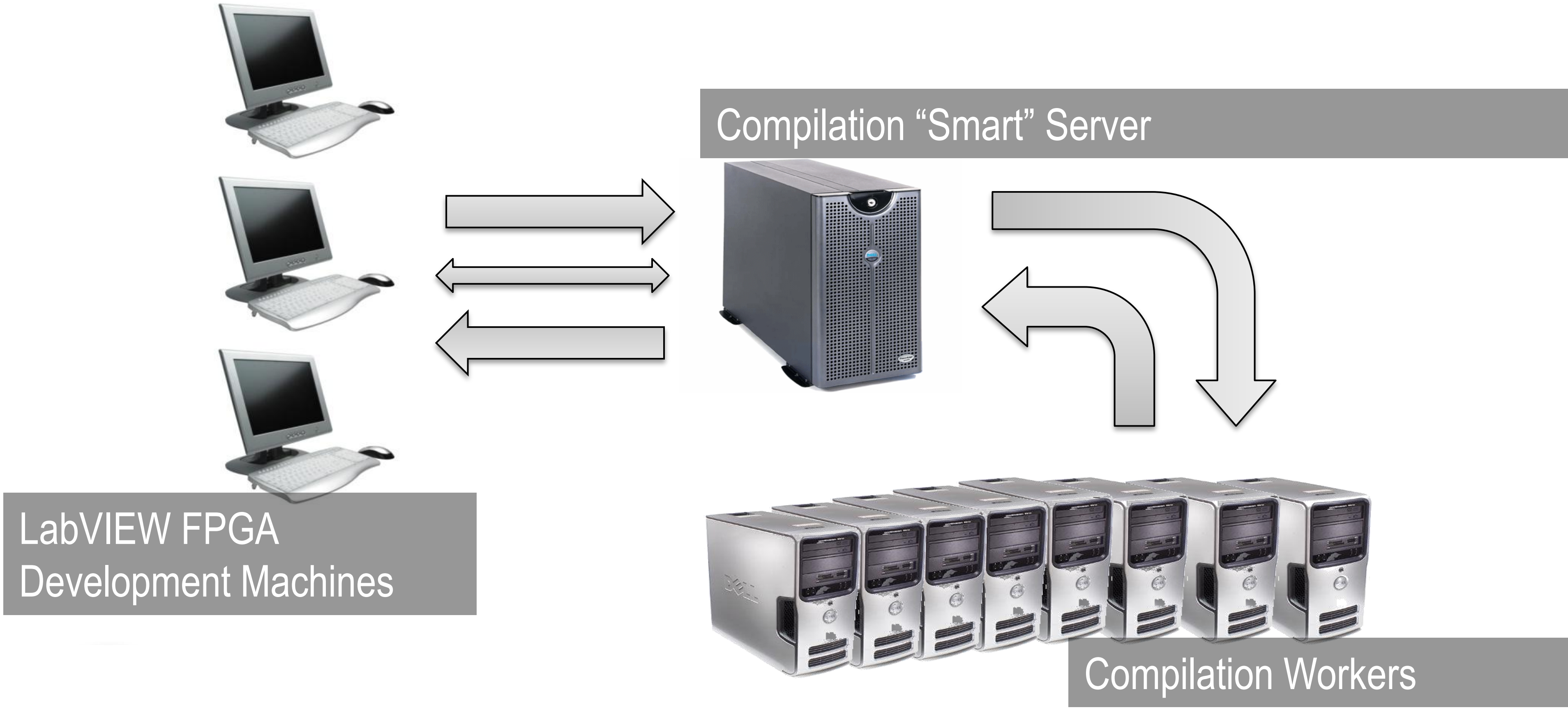
User name

Administrator

Password

****

☑ Log in automatically to this compile serv



Compilation "Smart" Server

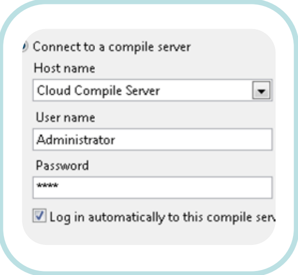
LabVIEW FPGA Development Machines

Compilation Workers

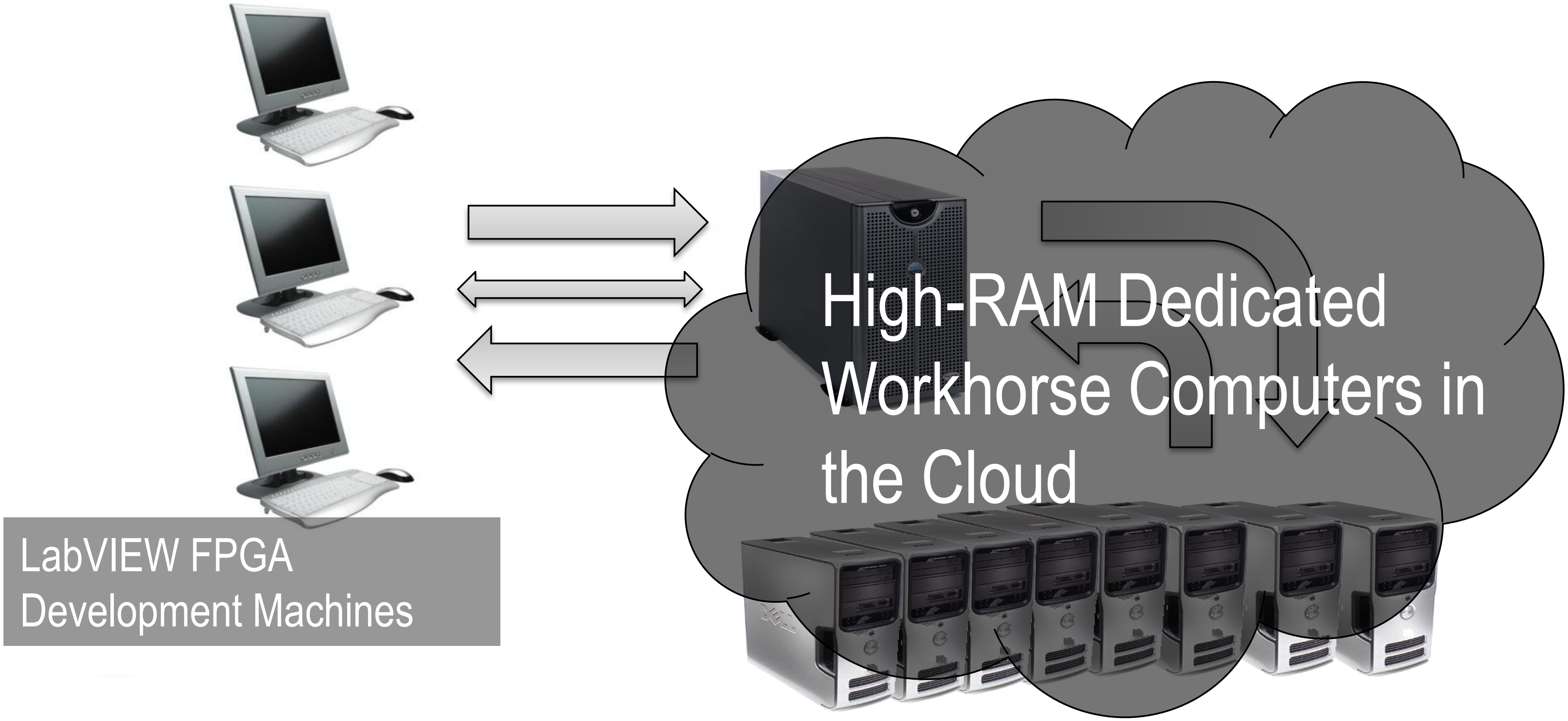
LabVIEW 2010 FPGA Compilation

LabVIEW FPGA Compile Cloud Service (Beta)

rick.kuhlman@ni.com



Connect to a compile server
Host name
Cloud Compile Server
User name
Administrator
Password
****
☑ Log in automatically to this compile serv



LabVIEW FPGA Development Machines

High-RAM Dedicated Workhorse Computers in the Cloud

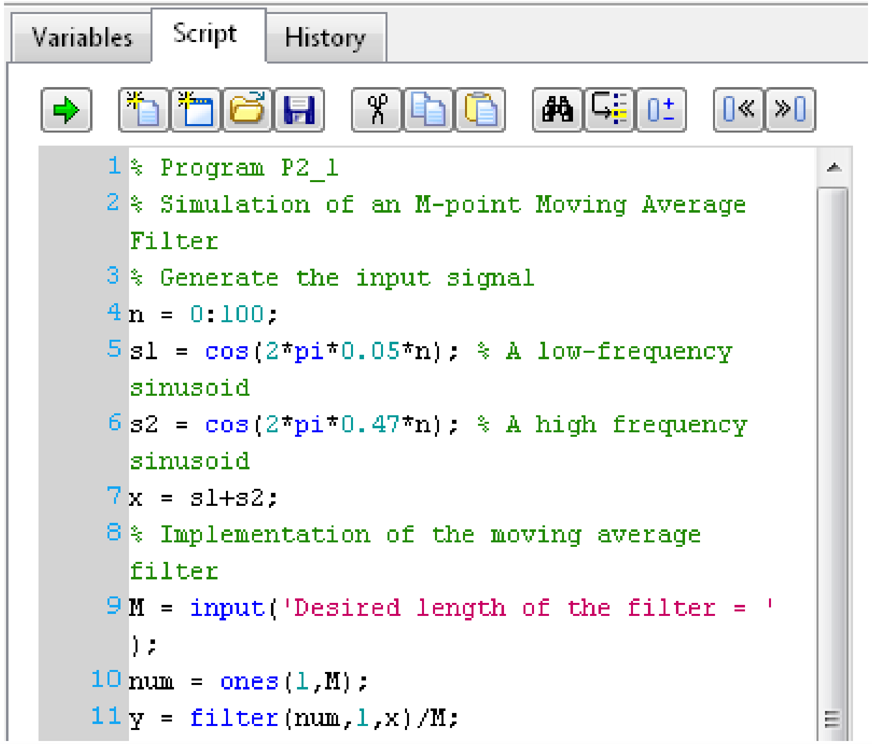
LabVIEW 2010 MathScript RT Module

MathScript Node

- Validate your custom .m files for deterministic behavior
- Automatically create output variables

MathScript Window

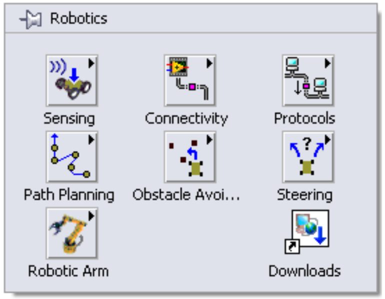
- Performance improvements
- Enhanced text-editor
  - Syntax highlighting
  - Line numbers
  - Find/replace text dialog box
  - Bookmarks



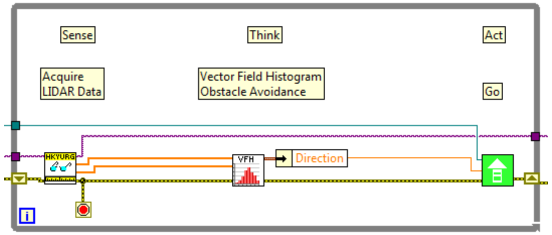
```
% Program P2_1
% Simulation of an M-point Moving Average Filter
% Generate the input signal
n = 0:100;
s1 = cos(2*pi*0.05*n); % A low-frequency sinusoid
s2 = cos(2*pi*0.47*n); % A high frequency sinusoid
x = s1+s2;
% Implementation of the moving average filter
M = input('Desired length of the filter = ');
num = ones(1,M);
y = filter(num,1,x)/M;
```

LabVIEW 2010 Robotics Module

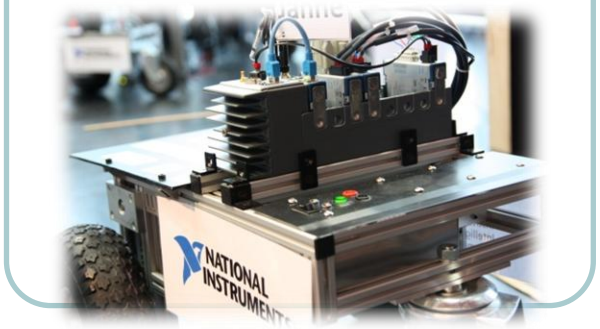
IP for sensing, navigation, motion control and more



High-level **graphical programming** environment

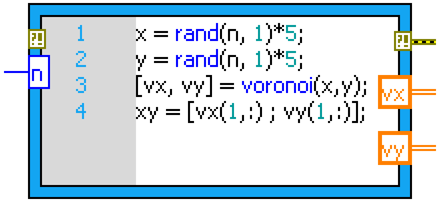


Deployment to Real-Time and FPGA hardware



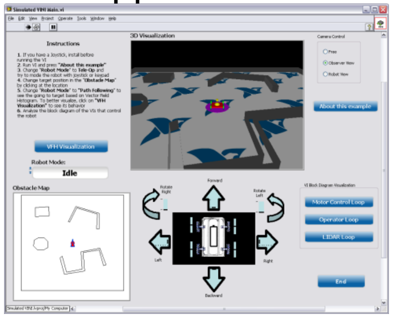
Connectivity to sensors and actuators from top vendors

Tools for integrating **text-based algorithms**



```
x = rand(n, 1)*5;
y = rand(n, 1)*5;
[vx, vy] = voronoi(x,y);
xy = [vx(1,:) ; vy(1,:)];
```

Examples of real-world applications



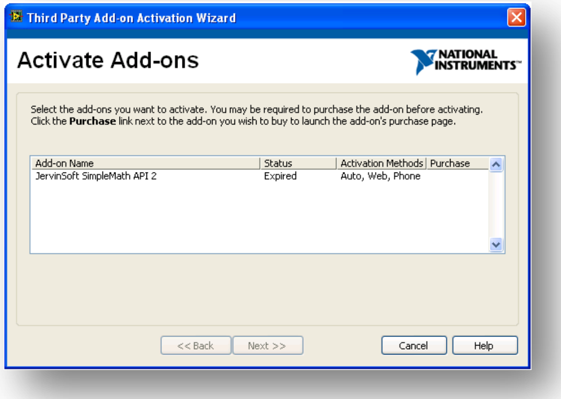
BUILDING LABVIEW ADD-ONS

LabVIEW 2010

Extending the Platform

Licensing and Activation for 3rd Party Add-ons

- Commercial Grade Activation solution from Concept Software
- Implement 30-day software trials for LabVIEW Add-ons you create
- Fully integrated in LabVIEW 2010

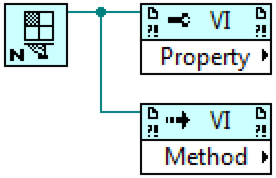


Select the add-ons you want to activate. You may be required to purchase the add-on before activating. Click the **Purchase** link next to the add-on you wish to buy to launch the add-on's purchase page.

Add-on Name	Status	Activation Methods	Purchase
JervinSoft SimpleMath API 2	Expired	Auto, Web, Phone	

LabVIEW Scripting

- Intended for power users to enhance the capabilities of LabVIEW during editing
- Used to inspect, modify, or generate LabVIEW code automatically



LabVIEW 2010 Resources

- LabVIEW 2010 New Features
- LabVIEW 2010 Performance Update
- LabVIEW Compiler: Under the Hood
- Timing and Synchronization in LabVIEW