VI Analyzer: The unsung hero of software quality

Chris Roebuck
Software Architect, Konrad Technologies
Chris Roebuck
Software Architect
Physicist
Team Working “Idealist”
LabVIEW Cheerleader
Vote with your feet
Credit where it’s due

• I was a CLA who did not use VI Analyzer
• I was a VI Scripting enthusiast
  • Inspired by Darren Nattinger’s quest to automate more of the SW process
  • Inspired by the great 3rd party tests at VI Analyzer Enthusiast community page

“I am neither clever nor especially gifted. I am only very, very curious.”
-Albert Einstein
What is software quality?

• Functional Quality
  Software functional quality reflects how well it complies with or conforms to a given design, based on functional requirements or specifications

• Structural Quality
  Software structural quality reflects how well it complies with non-functional requirements such as maintainability, scalability and robustness

Is it Fit for purpose?

Low quality software can cost money and time, now and throughout its life
Why does software have bugs?

A **software bug** is an error, flaw, failure or fault in a computer program or system that causes it to produce an incorrect or unexpected result, or to behave in unintended ways.

Software in general is created by humans:

• **Errors in coding (syntax etc)**
• **Misunderstanding of the required behavior**
• **Lack of understanding of environmental variables (OS, drivers etc)**
• **Faults in the development tool chain**
‘G’ things affecting quality / causing bugs

• Stylistic Issues
• Performance Issues
• Dangerous Constructs
To Improve Software Quality

• Do You Have a Spec?
• Do You Have Testers and or Unit Tests?
• Do You Hold Code Reviews?
• Do You Have a Build System (CI)?
• Do You Perform Usability Testing?
Code Linting

“Lint, or a linter, is a tool that analyzes source code to flag programming errors, bugs, stylistic errors, and dangerous constructs.”

- PC Lint
- ReShaper
- JSHint
- VI Analyzer

All these tools integrate into an IDE To some degree
VI Analyzer

• VI Analyzer performs static code analysis meaning it analyzes non-running (Idle) code

VI Analyzer inspects
• VI Properties
• Block Diagram
• Front Panel

VI Analyzer identifies
• Performance Issues
• Dangerous Constructs
• Stylistic Issues
What is the VI Analyzer?

Let's get started with VI Analyzer
Performance Issues
Stylistic Errors
Stylistic Checking

Icon View or Terminal View?
Label long wires where possible
Class Accessor Names
Front Panel Layout matches Conn Pane
(Potentially) Dangerous Constructs

- **User Cancelled or Error (43)**  
  Which did you handle?

- **Might behave differently in a built application**

- **Popup at run time if no file path provided**

- **Un-sanitized inputs into SQL statements**

- **Risk of default value causing downstream errors**

- **This will hang your system**

- **Empty refnum if Array is empty**
Why don’t developers use it?

• Show of hands ??

• “I do acceptance testing” “Customers don’t pay for style” “It fails for things I don’t care about” “It’s disruptive to the way I work” “There are so many results”

• Noise
• False Positives
• Workflow
Create a CFG

• Select what should be analyzed
• Select how it should be analyzed (which tests)

• Create different CFGs for different scenarios
  • Demo code
  • Everyday checking of code
  • Pre-delivery inspection
  • High Performance Applications
  • Cross platform Portability
Create and Save VI Analyzer Configurations
Style Guidelines

• Define your own rules
  • VI Properties (Filenames, Options)
  • Block Diagram
  • Front Panel

• Useful to get alignment of style within teams
Apply your own rules – Build Your Own Tests

- The VI Analyzer **Test Creator** provides a template for creating your own tests
- Use VI Server methods and properties to analyze code
- Put your tests in `<LABVIEW>project\_VI Analyzer\_tests\<YOUR DIR>\`
Roll your own

Identify array indicators that are not contained in a diagram disable structure, not on the conn pane either

- Array
  - 0
  - 0
  - 0
- Array 2
  - 0
  - 0
  - 0
- error out
- status
- code: 0
- source

- Ctl
  - IsOnConPane

- ControlTerminal

- Fails or needs repair

- Arrays in diagram disable

- Result Message
  - Ref Index

- Repair ?
Roll your own
FREE STUFF

VI Analyzer Enthusiasts group on NI.com
Chose to ignore some code

Check for #VIA_Skip bookmark in VIA test
VI Scripting

• Anything you can do with VI Scripting can be incorporated into a VI Analyzer test
• Scripting allows references to block diagram objects to be obtained
• Scripting allows the creation, removal and changing of items in a VI
Modify your failing code
There’s an API also!
Lets play with the VI Analyzer API
Check your code on each commit

“In general, the longer you wait before fixing a bug, the costlier (in time and money) it is to fix.” That guy Joel again !!
Check what you're committing.
Continuous Integration

Atlassian Bamboo Build Server & LabVIEW
Summary

• Reviewing Code Increases Quality (Performance, bugs, Style etc)
• VI Analyzer can perform your code review
• VI Analyzer can modify your code
• VI Analyzer can be integrated into your workflow
• Write a style guide
• Do it now!
Further Reading


Darren Nattinger – VI Analyzer Enthusiasts Community & Blog
Free later this week?

• JIRA Integration with LabVIEW  
  Wednesday May 22nd 2:30pm 12A

• Stop Breaking The Trunk - A Step By Step CI Tutorial for Test Programs  
  Thursday May 23rd 9:45am Room 12A

• Better, Faster, Stronger – It’s not all technical  
  Thursday May 23rd 11:00am Room 12B
Questions?

c.roebuck@Konrad-technologies.co.uk