Advanced Debugging with LabVIEW Task Manager

Submission No:

195

Submission Type:

Technical Session

Authors:

T. Vargo[1]

Speaker:

Timothy Vargo - Contact Me
Sandia National Laboratories

Description:

We will discuss both the merits and limitations of existing debugging tools in LabVIEW, explore which situations impede troubleshooting, then see a live demonstration of the free, open source and community developed "LabVIEW Task Manager"; which delivers new comprehensions into your running code, and interacting with individual or groups of VIs.

Learning Objective:

Learn new and better ways to troubleshoot and debug LabVIEW code, with a focus on a free debugging tool from the LabVIEW Advanced Virtual Architects (LAVA) user community -- the LabVIEW Task Manager.

Background Information:

** Motivation **
Like other major software development environments, the LabVIEW® IDE has long included a variety of debugging tools to help developers troubleshoot their own code. In addition to the standard debugging tools expected of any decent software development environment, such as Breakpoints, Single Stepping, and [Conditional] Probes (commonly called "Watch Variables" and "Watchpoints" in other languages), the graphical nature of LabVIEW also includes graphical-code specific debugging facilities, in the form of Execution Highlighting. Introduced with LabVIEW 2013, the Event Inspector added an ability to gain new understandings into the inner workings of our event structures. While all of these built-in tools are extremely useful during troubleshooting of individual VIs or small collections of VIs, particularly when we already know which VIs are the troublesome ones, the LabVIEW® IDE still lacks a debugging tool to provide insights with a "bigger picture" view of an entire project. VI Analyzer does its job well, but it intends to perform a static analysis of VIs, not a dynamic analysis of code while running. The separate Desktop Execution Trace Toolkit (DETT) product is also very good at what it was designed to do – but while dynamic execution traces can provide a fantastic wealth of evidence when event timing and/or sequence information is critical, they were never meant to provide other

(non-event) kinds of comprehensive awareness into the instantaneous state of your dynamic code.
Using only the native troubleshooting tools, reentrant VIs are especially difficult to debug, since each preallocated clone will have its own data space in memory. Dynamically launched VIs, even if not reentrant, also present their own particular frustrations during a debug session, since the developer is unable to place probes BEFORE executing the code. How to defeat these shortcomings, and many other debugging related challenges, have long been the topics of many user community discussions; and indeed several ad-hoc solutions and processes, although usually quite limited in scope, have been developed over the years by the community – but they lack the coherence of a unified tool.

** Solution **

The LabVIEW Task Manager seeks to be that missing unified debugging tool – to provide a dynamic & big-picture view of all VIs currently in memory, conquering those difficulties concerning reentrancy, clones, dynamic launching, finding & aborting hung VIs, and other sticky complications. LabVIEW Task Manager delivers new comprehensions into your running code, and enables interacting with individual or groups of VIs in many various ways, providing significant benefits while troubleshooting.

** History **

In September of 2011, a few community members discussed the need to create such a tool. Only a few days later a fledgling implementation was put forth by one very active community member, and then enhanced only two days after that by yet another highly respected community leader. By the end of only one week several others had joined the lively discussion, with a couple of them even contributing code modules which were then rolled into the tool by the original author. Seven months later the tool received improvements to error handling.
In October 2012, the original author released a plugin which allowed LabVIEW Task Manager to be used from inside a built executable. LabVIEW 2013 included an under-the-hood change which broke the tool's ability to detect some kinds of clones. This was fixed in July of 2014, along with the addition of yet a few more modest improvements and bug fixes; plus the tool was at that time packaged up using VI Package Manager (VIPM) for easier distribution, installation, and launching.

** What Is the LabVIEW Task Manager (LVTM)? **

LabVIEW Task Manager is a debugging tool for use during LabVIEW® code development. An expandable/collapsible tree diagram displays detailed information (both static and dynamic) on all VIs in memory, belonging to a selected project/target. It allows for interacting with single or multiple selected VIs at a time.

LVTMWhitePaper.docx

·Not the paper itself, but includes additional info and screenshots

Track and Summit Categories:

Software Development Techniques

Products Featured:

LabVIEW|