

文件类型: 仪器新闻简报
是否NI支持: 是
发布日期: Mar 22, 2012

适用于每个 LabVIEW开发 者的巧妙调试技巧

目录

1. 成为LabVIEW 漏洞修复高手
2. 利用探针
3. 调用时挂起
4. 未连线的前面板控件 和显示件
5. 执行跟踪
6. 单元测试
7. 有需要补充的内容 吗？

成为LabVIEW 漏洞修复高手

如果您经常使用 NI LabVIEW软 件，当软件不能按照 预期的情况运行时，您可能会有一个自己 的调试技巧和技术工 具箱。这些技巧和 技术有些是常用的，而剩下的可能是您自 己摸索出来的。

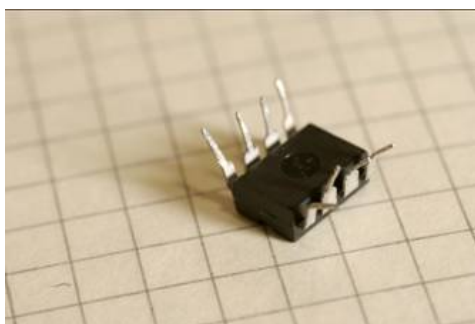


图1.巧妙的调试方法能够 清除漏洞。(Brian Searle摄)

我们与JKI的 LabVIEW程序 架构师 (Certified LabVIEW Architects, CLA)以及 LabVIEW社区 的朋友进行了交流，向他们学习一些快速 识别、隔离和修复漏 洞的“绝密”技巧。我们了解到以下几个 巧妙的调试技巧。

利用探针

探针是 LabVIEW调试 的第一个也是最简单 的工具，但它们却是 非常强大的工具。您 可极其轻松地 将探针 置于线路上，在查找 问题过程中不断获取 信息。以下是探针 的几个功能:

保存连线值功能

LabVIEW开发 环境中的保存连线值 功能经常会被忽 略。当您启用某个 VI的连线值保存功 能时，LabVIEW将自 动存储该VI框图上 每根连线的最后一个 值。然后，将光标 悬停在任一连线上，即使该VI未运行， 探针工具也将出现提 示信，显示该连线最 后一个值。

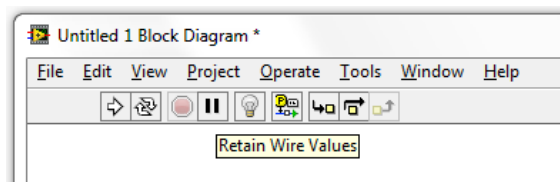


图2.保存连线值按钮就在 “灯泡”图标旁边。

自定义探针，无需编 写任何代码

运用LabVIEW 自定义探针，您无需 编写任何代码就可创 建强大、复杂的调试 工具。例如，您可在“自定义探针”>>“控件”>>“波形图”中创建显 示任意数字连线的先 前值的“历史探 针”。这可帮助您 发现您正在探究的数 据的模式或趋势，而 无需编写特殊的探针 代码。

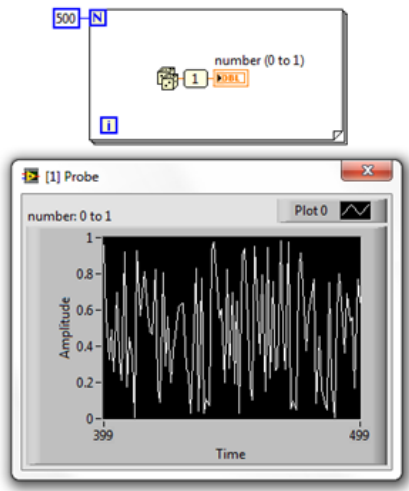


图3. 使用波形图探针显示 简单数字连线 的历史 值。

从LabVIEW工 具网络获取自定义探 针

借用他人创建的探针 来扩展您的工具 箱。SAPHIR 在LabVIEW工 具网络上提供了一系 列免费的自定义 LabVIEW探 针。您只需下载安 装VI软件包管理 器 (VIPM)社 区版 (也是免费 的)、搜索 ViBox ,再通过 几次单击就可安 装探 针。

如果您正在使用 LabVIEW 2011 ,您还可通 过单击 LabVIEW入门 指南窗口 的“查找 LabVIEW附加 工具...”从 LabVIEW工具 网络上获取 ViBox。

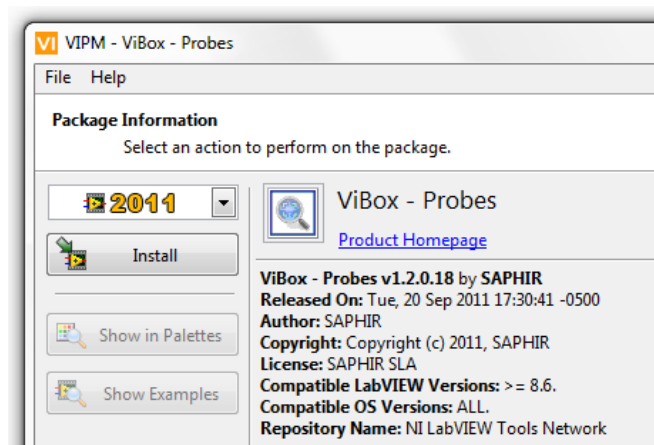


图4.使用VIPM安 装 SAPHIR自定义 探 针。

调用时挂起

许多LabVIEW 程序架构师甚至不 知道LabVIEW具 备该项功能。“调用 时挂起”功能可将执 行挂起 ,使您可以进 行以下任一或所有操 作 :

1. 在VI运行之前随意 更改VI的控件值。
2. VI返回调用程序时 随意更改显示件 的 值。
3. 重复运行VI,而无 需将执行返回至调用 程序。
4. 完全跳过VI执行 , 返回至调用VI。

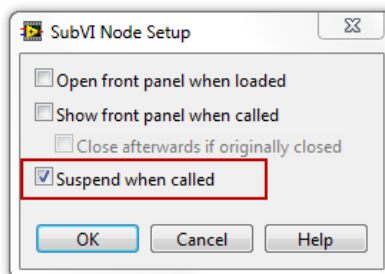


图 5. 在子VI节点设置窗 口中 ,选择“调用时 挂起”功能。

对于调用时挂起功 能,NI做了简要说 明,见 [此处](#)。然而 ,熟悉该功 能的最佳方法还是亲 自动手实践。

未连线的前面板控件 和显示件

为FIRST（崇尚发扬科学技术）机器人竞赛团队推荐了一项技术，这项技术对于专业的LabVIEW开发人员也非常有用

Ben建议在VI中添加前面板控件或显示控件，但并不与连线板连接。这些控件和显示件与“调试”输入或输出工作方式类似，您可在测试时对其进行手动控制。这可使您更容易在应用中生成偶然或极端事件，而无需繁多的支持或仿真代码的写入。

您还可以在已建应用程序中使用这些调试控件和显示件，前提是LabVIEW应用程序生成器包含了VI前面板。

[参阅 LVMastery.com上的介绍](#)，进一步了解Ben建议的调试子VI。

执行跟踪

如果其他方法都失败的情况下，有时您所能做的只有逐步跟踪代码。遇到此类问题时，您可通过几种方法在LabVIEW内进行执行跟踪。

最明显的方法是使用LabVIEW中内置的“高亮显示执行”功能，也称为“灯泡”。如果说探针是LabVIEW调试的首要工具，灯泡则位列第二。但是，如果碰到复杂或是时间相关的漏洞，您则可选择其他更有效的工具。

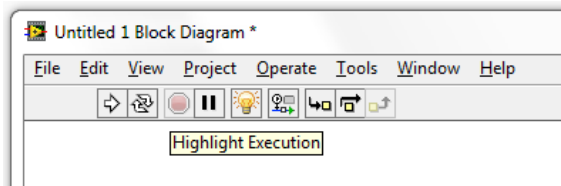


图6. 执行跟踪功能比灯泡功能更多。

跟踪磁盘或内存

启动执行跟踪的一个简单方法是在其中一个子VI中记录一个连线值。您可编写一个简单的VI，将该值记录到文件中，或甚至可创建一个自定义探针来显示连线的历史值。通常通过这样一个简单的跟踪您就可收集足够的信息来帮助您找到漏洞的原因。

NI LabVIEW桌面执行跟踪工具包

如果要调试一个大型的高度并行应用程序，则包含NI开发者套件2011的LabVIEW桌面执行跟踪工具包就是您进行LabVIEW代码分析的“利器”。它可提供关于正在运行的应用程序的各种信息，包括详细的内存分配和再分配、事件生成、由代码定义的自定义信息。Austin LabVIEW程序架构师论坛的创建人之一Fabiola De la Cueva指出该工具包可用于调试已崩溃的应用程序，因为它是独立于LabVIEW之外运行的。

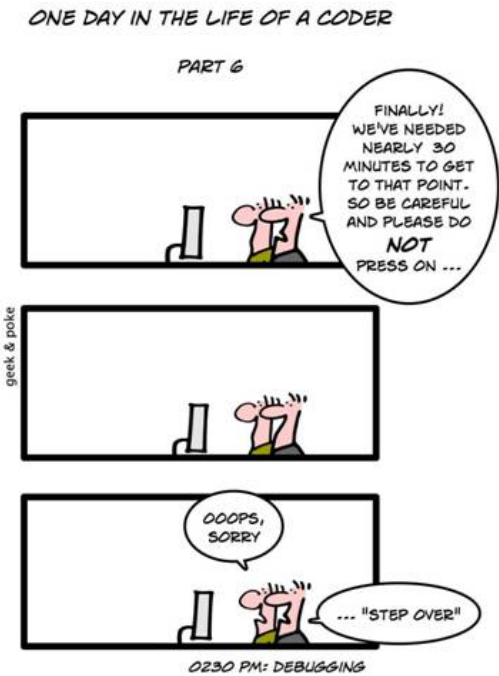


图7. 手动执行跟踪具有风险（图片来自 Oliver Widder）

[参阅教程](#)，进一步了解工具包。

专业的LabVIEW开发人员很庆幸，因为LabVIEW桌面执行跟踪工具包已包含在NI开发者套件2011之中，不会产生额外的费用。

单元测试

测试驱动开发 (TDD) 是一种 软件工程技术，可帮 助您从头创建更好的 代码，同时它也是帮 助您调试现有代码的 一个有效工具。

在TDD中，当您 发现应用程序存在漏 洞时，调试流程的 第一个步骤就是创建 一个可重现漏洞并导 致 测试失败的单元测 试。只有到那时 候，您才确实可以开 始尝试修复问题。如果单元测试通过， 则表明您已清除了该 漏洞。



图8. 你在哪里测试代码？

这种漏洞修复方法需 要费点时间，但是相 比其他较不精确的方 法，它具有以下几个 优点：

- 1. 单元测试具有可追溯 性，可完整显示漏洞 的发生过程以及您如 何将漏洞隔离
- 2. 您或其他开发者可随 时轻松验证或重新验 证您的修复，而无需 记住激活漏洞的详细 信息。
- 3. 单元测试可在您的项 目中与其他测试一起 自动执行，因此漏洞 如果再次出现，您可 以立即知晓。

针对单元测试有两个 主要的 LabVIEW附加 工具，TDD方法可 以与另一个一同使 用。 查看[JKI的免费VI测 试仪LabVIEW 附加工具](#)或参阅Eli Kerry的[LabVIEW软件 工程指南](#)，进一步解 LabVIEW单元 测试框架工具包，它 已包含在NI开发者 套件之内。

有需要补充的内容 吗？

无论您是经验丰富的 LabVIEW程序 架构师还是初出茅庐 的NI LabVIEW开发 工程师，你都可以学 习新技术来快速识 别、隔离和修复代码 中的漏洞（以及其他 代码）。现在，让 我们听听你的建议 吧。

您有独特的 LabVIEW使用 建议或技巧，可以让 您的同事大吃一惊 吗？访问[JKI论坛](#)与大家分享吧。

Omar Mussa是NI联盟伙伴 JKI的程序架构师 和专业服务经理。他 热衷于进行以目标 为导向的编程和测试 驱动开发。

Justin Goeres是NI联盟伙伴 JKI的程序架构师 和产品市场经理。虽 然已有14年的 LabVIEW使用 经验，但他仍喜欢每 天学习一些新的东 西，了解其他开发者 的观点。

此文首次发表于 2012年第一季度[仪器仪表季刊](#)

法律条款

资料受美国和其它国 家版权法的保护，禁 止任何违反版权法使 用该资料的行为，包 括但不限于重印、下 载、复制、改编，以 及通过任何媒体、设 备或过程的传播或传 送。