



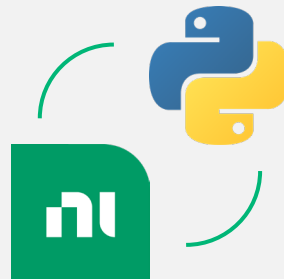
强强联合： Python与Test Workflow 相得益彰

The image displays a screenshot of the NI TestStand environment. On the left, the 'Steps' pane shows a test sequence with steps like 'Setup', 'Simulation Dialog', and 'Power On Test'. The main window shows a Python script titled 'EnclosedArea.py' with the following code:

```
1 #
2 -- Python-Script-File
3 -- Name
4 -- Author : National Instruments Ireland Resources Limited
5 -- Comment
6
7
8 import DIAdem
9 ds=DIAdem.Application
10
11 if False:
12     import DIAdem_CodeCompletion as dd
13
14 #Path documents = dd.ProgramDir + "Examples\Documentation\
15 #Path data = dd.ProgramDir + "Example\Unfa1\"
16
17 -----
18 -----
19 -----
20 -----
21 -----
22 -----
23 # - Create report
24 # - resultGroupChns : Channel group containing the results of all area calculations
25 # - chnsResultSegments : Calculated segments (intersections) of all curves
26 # - chnsAreaGroupChns : Channel group containing the results of all area calculations
27 # - chnsArea : List of all x-channels that were used for the area calculation
28 # - chnsAreaY : List of all y-channels that were used for the area calculation
29
30 # Create report page (AppendLayoutPage, chnsResultGroupChns, chnsResultSegments, chnsAreaGroupChns, chnsArea, chnsAreaY)
31 # Append layout when necessary
32 ds.Report.AppendLayout (AppendLayoutPage)
33 ds.Report.AppendLayout (AppendLayoutPage)
34
35 # Create sheet
36 ds.Report.Sheets (dd.Report.Sheets.Count)
37 ds.Report.Sheet.Name = "Intersection points" + ds.ResultGroupChns(1).Size - 2)
38 ds.Report.Sheet.Channels (["Area", "AreaY"])
39
40 # Add filled area
41 filledArea = ds.Channels.Add (ds.Channels.FilledArea, "D0FilledArea")
42 filledArea.Shape.XChannel.Reference = chnsAreaGroupChns(1).Name).getReference (ds.ExtendedChannelName)
43 filledArea.Shape.YChannel.Reference = chnsAreaGroupChns(2).Name).getReference (ds.ExtendedChannelName)
```

On the right, the 'Data Portal: Internal Data' pane shows a table of data with columns for Name, Description, Index, and Source name. The table includes entries for 'Example', 'Noise_1', 'Noise_2', 'Noise_3', 'Noise_4', and 'Noise_5'. The 'Base Properties' section shows the following details:

Name	Example
Description	DIAdem standard ex
Index	1
Source name	Example



工作效率是保持领先优势并确保按时交付项目的关键。您肯定希望使用最适合的工具来完成任务，比如总是可以使用所选的编程语言来构建所需的内容，或者使用现成即用的软件来减少开发工作。Python是一种通用编程语言，因其渐进式学习曲线、零使用成本和强大的数据分析能力而广受欢迎。Test Workflow是NI的一款热门测试软件套件，囊括仪器自动化、测试序列生成、数据分析等专用工具！

对于这两者，您并非只能择一而选。Test Workflow软件非常灵活，可以与Python等其他编程语言结合使用。利用NI现成即用的软件，使用预先构建好的功能，节省开发时间，同时也可使用Python进行高级定制和实现自动化。

利用现有的Python代码

采用Test Workflow并不意味着从零开始。您完全无需浪费时间重新开发Python代码。为了提高效率，您可以使用每种工具最强大的功能并复用现成的功能。您可以复用现有程序将它们集成到LabVIEW和TestStand中，或者使用G Web开发软件为Python程序构建Web界面。

在LabVIEW中调用Python函数

LabVIEW是一种图形化编程环境，用于帮助工程师开发自动化测试系统。其强大之处在于可实现仪器的自动化，并通过内置的用户界面来监测测试。在LabVIEW 2018中，NI引入了Python节点，使用户能够在执行LabVIEW程序时调用Python文件。譬如，您想利用另一位同事使用Python编写的现有数据分析库，同时利用LabVIEW出色的仪器连接性。您可以只使用Python，但在连接到硬件时可能会遇到问题。这时，LabVIEW中引入的Python节点就为两种语言之间的互操作提供了有效的方法。

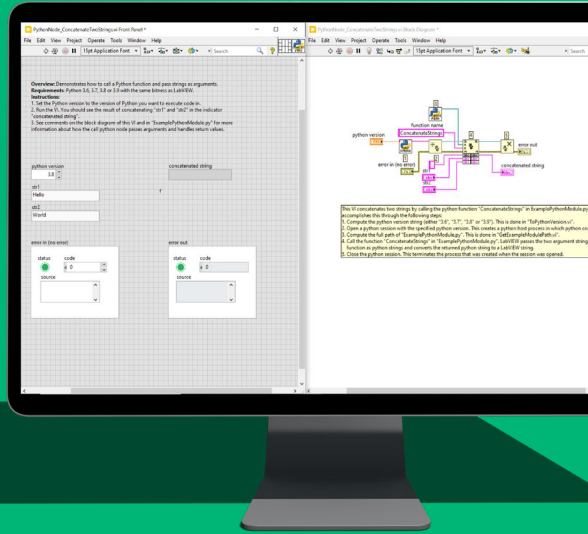


图1：显示器上显示了LabVIEW正在调用一个Python脚本。

TestStand中的Python代码序列生成

TestStand是工程师用于开发自动化测试序列的测试执行软件。它提供了一个用于创建测试序列的交互式开发环境，以及测试执行程序的所有功能：单元跟踪、报表生成、并行测试执行等。借助TestStand Python Adapter，用户可以调用Python代码模块，将其作为测试序列的一部分。虽然您也可以使用Python来构建测试执行程序，但这需要花费大量的时间，而TestStand恰好可提供此功能。因此，将两者的优势相结合，您就可以在TestStand中复用所有现有的测试代码，而无需从零开始。

“TestStand提供了一种现成即用的解决方案，诸多关键特性有助于缩短开发时间，而LabVIEW提供的本地功能则可帮助控制必要的测试系统硬件。”

- Dillon Glissmann
DISTek Integration, Inc.

基于G Web远程访问Python测试

G Web开发软件是一个图形化编程环境，专为开发基于Web的测试用户界面而优化。虽然很多人已经是技术娴熟的编程人员，但这些技能并不一定能够满足使用HTML或JavaScript进行Web开发的需求。这正是G Web的最强大之处，因为它是一个交互式环境，提供了用于数据显示和用户控制的预构建对象。对于使用Python编写的现有测试应用程序，G Web包含可简化数据传输和通信的预打包API。无需具备高级Web开发知识即可将Python测试系统部署到Web。

使用Python控制NI软件

技能熟练的Python编程人员可以扩展Test Workflow的功能来实现高级自动化。使用FlexLogger和DIAdem的内置功能来简化测试设置和数据分析，然后迁移到Python来实现未来流程的自动化。

FlexLogger中的自动化数据采集

FlexLogger是一款无代码应用软件，用于结合NI DAQ硬件执行数据采集。该应用程序可直观地设置测试和存储数据，从而缩短了首次测量的时间。虽然也可结合硬件使用NI提供的Python API执行测试，但FlexLogger提供了更高的灵活性，只需单击几下即可修改测试。利用FlexLogger中简化的设置，您可以使用FlexLogger Python API来实现自动化，进而提高工作效率。这可让您控制任务的执行并通过编程更改Python代码中的配置。

DIAdem中的脚本数据分析和报表生成

DIAdem是一款测量数据分析工具，用于查看和分析数据并生成报表。DIAdem包含了用于索引目录以查找数据的高级工具、用于分析的内置函数以及用于查看数据的图表和图形。利用DIAdem，您无需自己编写程序来创建复杂的可视化呈现和执行高级计算，其本身自带此类功能。您只需一款工具即可完成整个后期数据处理工作流程。此外，您还可使用Python编写的脚本实现从导入到报表生成整个过程的自动化。与其在不同的程序中构建分析库或生成报表，不如选择DIAdem，一步到位！

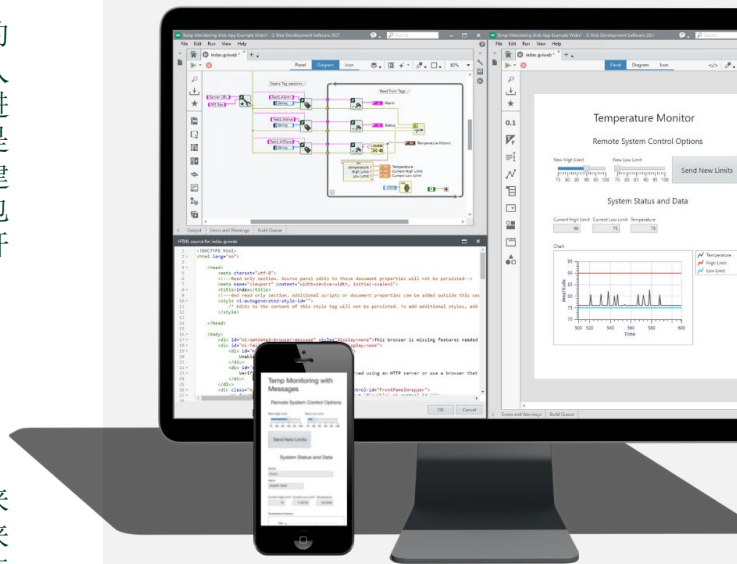


图2：显示器上显示了G Web开发软件，智能手机显示了Web应用程序。

“我们的报表生成和分析时间减少了95%，并成功地使用一键式DIAdem解决方案替代当前的多步骤流程。”

- Jim Knuff

Raytheon Missile Systems

©2022 NATIONAL INSTRUMENTS. 版权所有NATIONAL INSTRUMENTS、NI、NI.COM和LABVIEW均为NATIONAL INSTRUMENTS CORPORATION的商标。此处提及的其他产品和公司名称均为其各自公司的商标或商业名称。

[立即试用Test Workflow](#)