



Better Together: Python & Test Workflow

The screenshot displays the NI TestStand Sequence Editor interface. The main window shows a Python script titled 'EnclosedArea.py' with the following code:

```
1 # Python-Script-File
2 # -- Name : --
3 # -- Author : National Instruments Ireland Resource Limited
4 # -- Comment :
5
6
7
8 import DIAdem
9 dd=DIAdem.Application
10
11 if False:
12     import DIAdem_CodeCompletion as dd
13
14 #PathDocuments = dd.ProgramDir + "Examples\Automation\
15 #PathData = dd.ProgramDir + "Examples\Data\
16
17
18 # Begin Function Definition
19
20
21
22 # Create report
23 # append : Appendix for channel group name
24 # cbaResults : Channel group containing the results of all area calculations
25 # cbaResultSegments : Calculated segments (intersections) of all curves
26 # cbaResultGroupChans : Channel group containing the results of all area calculations
27 # cbaMax : List of all x-channels that were used for the area calculation
28 # cbaMaxY : List of all y-channels that were used for the area calculation
29
30 @data.ReportPage(appendLayoutPage, cbaResultGroupChans, cbaResultSegments, cbaResultGroupChans, cbaMax, cbaMaxY)
31 # Append layout when necessary
32 if appendLayoutPages
33     ds.Report.AppendLayout([PathDocuments + "EnclosedArea.TSR"])
34     ds.Report.AppendLayout([PathData + "EnclosedArea.TSR"])
35
36 # Create sheet
37 # ds.Report.Sheets(dd.Report.Sheets.Count)
38 # ds.Sheet.Name = "Intersection points: " + ds.AttributesResultGroupChans(1).Size - 2)
39 # ds.Sheet.Objects("DDUIS1").Curve2D
40
41 # Add filled area
42 # illicArea = <Curve2D.Add(ds.o2DShapeFilledArea, "FilledArea")
43 # illicArea.Shape_XChannel.Reference = cbaResultGroupChans("X_Max").GetReference(ds.ExtendedChannelName)
44 # illicArea.Shape_YChannel.Reference = cbaResultGroupChans("Maximum").GetReference(ds.ExtendedChannelName)
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

The right-hand side of the interface shows a 'Data Portal: Internal Data' window with a table of data points:

Channel Name	Value	Units
Time	s	
Speed	m/h	
RPM	1/min	
Torque	Nm	
Power	kW	
Noise_1	Pa	
Noise_2	Pa	
Noise_3	Pa	
Noise_4	Pa	
Noise_5	Pa	
Results_Noise data		
Room temperatures		
Temperature distribution		

Efficiency is the key for staying ahead and ensuring you meet project timelines. When completing tasks, you want to make sure you're using the best tool for the job. You can always build what you need in the programming language of your choice or there's off-the-shelf software, which reduces development. Python is a general-purpose programming language that is popular for its gradual learning curve, no cost of use, and strength in data analysis. And Test Workflow is a bundle of NI's most popular software for test, which includes purpose-built tools for instrument automation, test sequencing, data analysis, and more!

However, you don't only need to choose one. Test Workflow software is flexible with the ability to work with other programming languages, like Python. By using off-the-shelf software from NI, save development time with pre-built functionality and use Python for advanced customization and automation.

Leverage Existing Python Code

Adopting Test Workflow doesn't mean you need to start from scratch. Time spent redeveloping code in Python would be wasted. In aiming for efficiency, you want to use what each tool is best at and re-use what is readily available. You can re-use existing programs and integrate them into LabVIEW and TestStand or build a web interface for a Python program in G Web Development Software.

Call Python Libraries in LabVIEW

LabVIEW is a graphical programming environment engineers use to develop automated test systems. It excels at automating instruments and monitoring test with built-in user interfaces. In LabVIEW 2018, NI introduced the Python Node, which allows users to call a Python file in the execution of a LabVIEW program. For example, you want to leverage an existing data analysis library written in Python from another colleague, while taking advantage of LabVIEW's instrument connectivity. You could just use Python, but you would probably experience challenges when connecting to your hardware. This approach offers an efficient way to interoperate between languages.

Sequence Python Code in TestStand

TestStand is a test executive software that engineers use to develop automated test sequences. In TestStand, you have an interactive development environment for creating test sequences, as well as all the functionality of a test executive: unit tracking, reporting, parallel test execution, and more. With the TestStand Python Adapter, users can call Python code modules as part of their test sequences. While you could build a test executive in Python, you would have to spend countless hours and days building the functionality TestStand already has. By leveraging the strengths of both, you can re-use all your existing test code with TestStand without needing to start from scratch.

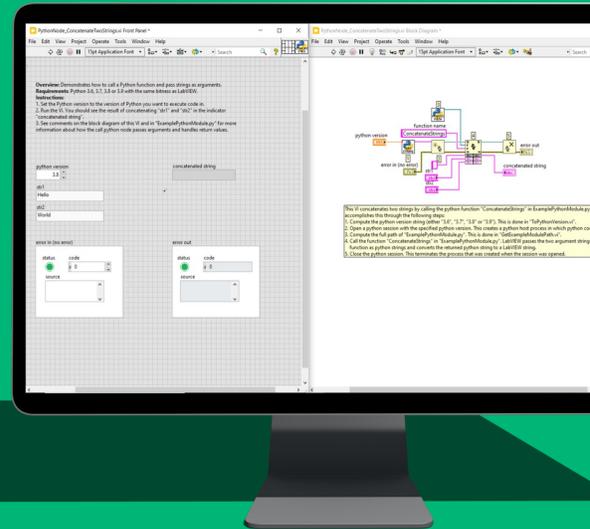


Fig. 1: Monitor with LabVIEW calling a Python script.

"TestStand provided an out-of-the-box solution that reduced development time by supplying a number of key features, while LabVIEW contained the native functionality for controlling the necessary test system hardware."

- Dillon Glissmann
DISTek Integration, Inc.



Access your Python Test Remotely with G Web

G Web Development Software is a graphical programming environment optimized for developing web-based user interfaces for test. While many of us are skilled programmers, those skills do not always align with the needs for web development in HTML or JavaScript. This is where leveraging G Web can be the most efficient as it is an interactive environment with pre-built objects for data display and user control. And for existing test applications written in Python, G Web includes pre-packaged APIs that simplify data transfer and communication. Make your Python test system web ready without advanced web development knowledge.

Control NI Software with Python

As a skilled Python programmer, you can expand functionality in Test Workflow for advanced levels of automation. Use the built-in functionality of FlexLogger and DIAdem for simplifying test set-up and data analysis – then move to Python to automate the process for the future.

Automate Data Acquisition in FlexLogger

FlexLogger is a no-code application software for performing data acquisition with NI DAQ hardware. This application reduces time to first measurement with an intuitive way to set up your test and store data. While NI provides Python APIs for performing test with our HW, FlexLogger enables you to be more agile and modify tests in a few clicks. By leveraging the simplified setup in FlexLogger, you can focus on efficiency through automation using FlexLogger Python APIs. This allows you to control the execution of tasks and programmatically change configuration in your Python code.

Script Data Analysis and Reporting in DIAdem

DIAdem is a measurement data analytics tool for viewing, analyzing, and reporting on data. DIAdem contains advanced tools for indexing your directory to find data, built-in functions for analysis, and charts and graphs for viewing data. Rather than writing your own program to create complex visualizations and perform advanced calculations, DIAdem contains this natively. You can complete your entire data post-processing workflow in one tool. Additionally, go even further to automate the entire process from import to report with scripts written in Python. Rather than building an analysis library or developing reports in disparate programs, go with DIAdem.

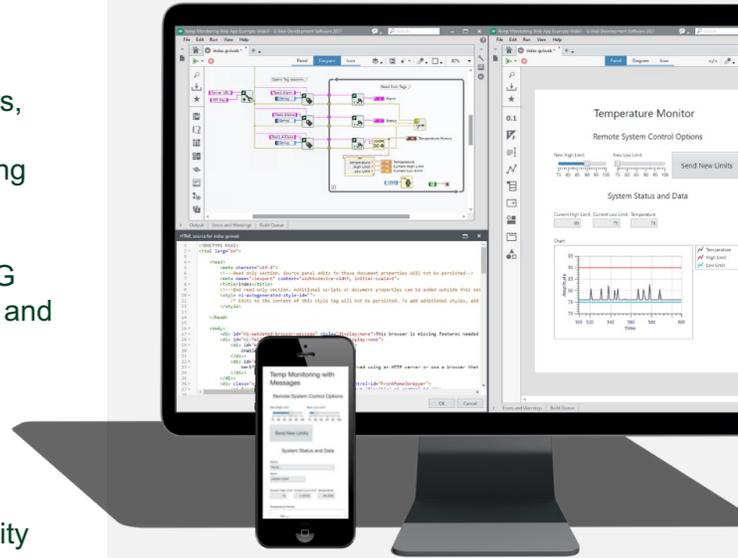


Fig. 2: Monitor with G Web Development Software and smartphone displaying web application.

"We have reduced our reporting and analysis time by 95 percent and achieved our goal of replacing the current multistep process with a one-button DIAdem solution."

- Jim Knuff
Raytheon Missile Systems

[Try Test Workflow Now](#)

©2022 NATIONAL INSTRUMENTS. ALL RIGHTS RESERVED. NATIONAL INSTRUMENTS, NI, NI.COM, AND LABVIEW ARE TRADEMARKS OF NATIONAL INSTRUMENTS CORPORATION. OTHER PRODUCT AND COMPANY NAMES LISTED ARE TRADEMARKS OR TRADE NAMES OF THEIR RESPECTIVE COMPANIES.

testforce

+1 (888) 880-6804
sales@testforce.com

ni.com