LABVIEW 2013 NEW FEATURE DEMO SCRIPT (WHAT'S NEW)

SUMMARY

This demo explores many of 2013's new features, including:

- LabVIEW Bookmark Manager
- <u>Attachable Comments</u>
- <u>New Excel Integration (Optional)</u>
- Mouse Wheel Support for Controls
- Event-Based Programming Improvements
 - Static Event for Mouse Wheel Interaction
 - o Event Inspector Window
 - o High Priority User Events
- <u>Simplified Web Service Experience</u>
 - o Project Item
 - o Debugging
 - o Deploy with Executable

DEMO REQUIREMENTS

- LabVIEW 2013
- Application Builder

SETUP INFORMATION (PRIOR TO PRESENTATION)

1. Download the What's New in LabVIEW 2013 VI Package file, and double-click on the VI package file.



2. Choose to "Install" the VI package file.

VI JKI VIPM - What's New in	LabVIEW 2013	x
File Help		
Package Information		
Select an action to	o perform on the package.	
12 2013 V	What's New in LabVIEW 2013	
	What's New in LabVIEW 2013 v1.0.0.4 by NI Author: miustiss	
Show in Palettes	Copyright: Copyright (c) 2013, NI Compatible LabVIEW Versions: >= 2013. Compatible OS Versions: ALL.	
Show Examples	Description: This package contains the necessary files to present the What's New in LabVIEW 2013 presentation and demo.	
*Browse All Versions		Ŧ

3. From the next dialog, choose "Continue".

1 7 6 11						
Select an action to perform on the package.						
± 2013 ▼	Here is a list of task	s that VIPM will perform. Click th	ne checkbox to enable	e or disable the action or	n the item.	
	Product /\	Action	Status			
Install	What's New in LabVIEW	2013 v1.0.0.4 to be installed	user selected			
	·					
🔍 Show in Palettes						
*						
Show Examples						
	Include Dependencies					

4. From the resulting dialog, choose "Finish".

NOTE: If Package Manager claims it cannot access LabVIEW over VI Server, open LabVIEW and Tools >> Options drop-down menu. Look at the "VI Server" category, and ensure both "Machine Access" and "Exported VIs" subcategories have "*" items. This will be absolutely sure that Package Manager may access LabVIEW.



5. From the LabVIEW Getting Started Screen, choose "Create Project".

LabVIEW	
File Operate Tools Help	
LabVIEW 2013	(Search Q)
Create Project	Open Existing
	Show Projects
What's New in LabVIEW 2013	Whats New in LabVIEW 2013.lvproj
What's New in LabVIEW 2013	DDEserver_Debug.lvproj
	C:\Users\mjustiss\Desktop\new_shit\new_shit\Main Whats New
	Untitled Project 14.Ivproj
	Untitled Project 13.lvproj
	Untitled Project 12.lvproj
v	C:\Users\miustiss\Desktop\package.for.whats.new.demo\Source
Find Drivers and Add-ons Connect to devices and expand the Participate in the discuss	upport ision forums or Leam to use LabVIEW and upgrade
functionality of LabVIEW. request technical support	t. from previous versions.

6. Under "Demonstrations", choose "What's New in LabVIEW 2013" and click "next"

hoose a starting point for the proj	ect:
All Templates Sample Projects CompactRIO Desktop NI-579X Demonstrations	What's New in LabVIEW 2013 Demonstrations Launches the What's New in 2013 Project and includes the presentation and demo script as supporting files. More Information

NOTE: Once the VI Package file has been installed. You only need to begin this presentation and demo from the "create project" dialog to start with the same files each time. All necessary files will be accessible from the project created through this dialog.

STEP-BY-STEP INSTRUCTIONS

- 1. DO: PRESENT THE WHAT'S NEW SLIDE DECK THROUGH THE DATA DASHBOARD 2.2 SLIDE.
- 2. Explain: Each year, NI strives to incorporate user feedback when determining the features we'll add to LabVIEW. Now that we've discussed some of the new hardware for 2013, I'd like to jump over to LabVIEW and show you some of the new features we've implemented to help you be more productive.
- 3. DO: Ensure the "Whats New in 2013" project is open.
- 4. DO: Open Main.VI and Run It
- **5.** Explain: Anyone who has ever inherited and had to interpret existing LabVIEW code knows the value of good documentation.
- 6. DO: Open the Bookmark example VI by clicking the "Bookmark Example" button on the Main.vi front panel.



7. Explain: Let's imagine we inherited this particular application, based on a LabVIEW sample project, and were tasked with adding additional data logging functionality. Where to start?

8. DO: Point out the "Logging Message Loop" VI, open it, and switch to its block diagram.



- 9. Explain: For many applications, our code and VI hierarchies quickly become so complex we cannot identify, at a glance, where specific functionality resides. In this case, where I want additional data logging functionality, I could continue to parse through this code, ultimately determining where modifications were necessary, but what if there was a better way? In LabVIEW 2013, there is, and it's called bookmarks.
- 10. DO: Click the "View" dropdown and select "Bookmark manager".



11. Explain: In 2013, I can turn any comment into a bookmark by simply beginning the comment with a hash tag (#). This prompts LabVIEW to recognize that comment as a bookmark, and aggregate it into this bookmark manager window.

12. DO: Highlight the bookmark titled "#1-Bookmark".

😰 Bookmark Manager		
Whats New in LabVIEW 2013.lvproj	● Group b ◎ Group b	oy Bookmark oy VI
Bookmarks	VIs	*
#1-Bookmark		
#1-Bookmark Modify Data Logging parameters in this VI	Log Data.vi	
#2-Attached_Comments		
#2-Attached_Comments Attachable comments show visual relationship and maintain relationship after block diagram cleanup.	Main.vi	E
#3-XLSX_Integration		
#3-XLSX_Integration Show the new write to .xlsx functionality in the Write To Measurement File Express VI	Log Data.vi	
#4-Mouse_Scroll_Events		
#4-Mouse_Scroll_Events Mouse wheel interaction added for controls and event structure	Main.vi	
#5-Event_Inspector_Window		
#5-Event_Inspector_Window Event Inspector Window helps debug	Main.vi	-
	Close	Help

- **13. Explain:** Notice I've left a to-do bookmark for, perhaps, another developer, indicating I want this parameter change made.
- 14. DO: Double-click on the "#1-Bookmark" bookmark.
- 15. Explain: By simply double-clicking this bookmark, I am able to navigate them directly to the location in my application I wished for them to view and modify. In LabVIEW 2013, all sample projects will make use of the bookmarks and this feature will help single and multi developer efforts alike by simplifying the process of code navigation and note aggregation. <u>NI considers this to be a best practice for code documentation moving forward.</u>

NOTE: You may not navigate to Bookmarks contained in running LabVIEW code. In this demo, the Main VI is running, but the sample project code, used to show Bookmarks, is not running to facilitate this demo.

- 16. Explain: Another feature we've implemented, to complement the addition of bookmarks, is the ability to attach comments on the block diagram. In past, LabVIEW developers used free labels to describe algorithms, leave placeholder notes, and generally put their thoughts onto block diagrams. But, aside from decorations, there has never been a way to create a direct and visual relationship between a comment and the code it describes. LabVIEW 2013 changes this by allowing us to attach our comments to any structure, function, or constant on the block diagram.
- **17. Explain:** To attach this comment to any function, structure, or constant on my block diagram, I simply move my mouse to the bottom right, click the arrow icon that appears and click again on the entity on my diagram I care about.
- **18.** DO: Move mouse to bottom-right of "#1-Bookmark" comment, look for yellow boxed arrow to appear, click and move mouse to show the carried arrow. Move mouse to the border of

the case structure and click after the structure is highlighted.



- **19. Explain:** Now that I've attached this comment, I may move the comment wherever I like and still maintain a visual relationship with this control I've chosen.
- **20.** DO: Click and move the comment around to show this relationship.



- **21. Explain:** Another great attribute of attached comments is their persistence through block diagram cleanup. After attaching, LabVIEW's block diagram cleanup utility will now take this attachment into account, and attempt to keep the comment close to the item it's attached to. No longer will documentation be sent adrift when using this convenient utility.
- **22.** DO: Close the "Log Data" VI, Bookmark Manager Window, and "Bookmark Example" VI. Save no changes. You should now be back to the front panel of Main.VI
- **23. Explain:** We'll use this project to show you some additional new productivity features of LabVIEW 2013. This project is designed to simulate a temperature alarming application, and is based on the common producer/consumer, event-based architecture.

OPTIONAL FEATURE DEMO

24. Do: Double-click Bookmark #3-XLSX_Integration to open the Log Data VI.

📴 Bo	ookmark Manager nats New in LabVIEW 2013.Ivproj 🗨		Group by Bookmark Group by VI
Book	marks	VIs	
=	#1-Bookmark		
	#1-Bookmark Modify Data Logging parameters in this VI	Log Data.vi	
-	#2-Attached_Comments		
	#2-Attached_Comments Attachable comments show visual relationship and maintain relationship after block diagram cleanum	Main.vi	E
	#3-XLSX Integration		
	#3-XLSX_Integration Show the new write to .xlsx functionality in the Write To Measurement File Express VI	Log Data.vi	
	#4-Mouse_Scroll_Events		
	#4-Mouse_Scroll_Events Mouse wheel interaction added for controls and event structure	Main.vi	
-	#5-Event_Inspector_Window		
	#5-Event_Inspector_Window Event_Inspector Window belos debug	Main.vi	-
			Close Help

25. Do: Double-click the "Write to Measurement File" Express VI



26. Explain: In LabVIEW 2013, you may now write data directly to an Excel file type from the Write to Measurement File Express VI. So, if all I need to do is take my data and export it, I have a quick way to do this using LabVIEW 2013, without needing to use the .lvm text file format.

NOTE: This functionality is WRITE-ONLY. We cannot read this file back using the Read From Measurement File Express VI. Also, this functionality makes use of the Open Office file format. So, once the resultant .xlsx file has been opened and saved in Microsoft Excel, the NI Report Generation Toolkit for Microsoft Office would then be necessary to use this file in LabVIEW. Moreover, no extensive formatting options are available, so the Report Generation Toolkit is still recommended for any report creation in LabVIEW.

27. Explain: Each year, we try to incorporate our user's feedback, and a highly requested feature has been native mouse wheel support for front panels. We've added this support in 2013, allowing simplified programming of mouse wheel interaction with LabVIEW UIs.

LABVIEW 2013 NEW FEATURE DEMO SCRIPT, Page 9 of 21

28. DO: Click the "Acquire" button to populate the temperature graph.



- **29. Explain:** In this example application, I'm performing some logic on my simulated temperature readings to determine if I should alarm my user, based on a limit they've set. [The default limit value is 65, which should alarm when you click "acquire"] If I want to quickly change my limit value, I can now accomplish that with my mouse wheel.
- **30.** DO: Hover the mouse over the "Temperature Limit" control and mouse wheel up and down to change the control value. This should also update the cursor value on the graph. [You may also use your touchpad's scroll if there is no mouse]



31. Explain: As I mentioned, this program captures UI interaction with events. In 2012, we introduced new templates and sample projects as starting points and references for best practice architectures. Many of these center around using Event Structures to capture user interaction. And, in adding mouse wheel support, we've added an event to capture any mouse wheel interaction with items on our front panel. This allows me to do things like better visualize the data I've taken by zooming in and out on my graph.

32. DO: Mouse wheel up and down on the "Temperature Graph". This should automatically adjust the X and Y scale to simulate a zoom-in and zoom-out effect.



- **33. Explain:** In this case, I've written a VI to programmatically adjust the scales of my graph. [Just to point out this is not built-in functionality] But, this mouse interaction could be captured to prompt any response your application requires.
- **34.** Explain: I mentioned templates and sample projects, and their use of event structures. We consider it best practice for any application, which must respond to user input, to leverage an event structure. In 2013, we're making it simpler to debug event-based programs. We have always been able to use standard LabVIEW debugging tools for events, but this wasn't always sufficient for applications processing a high volume of events.
- 35. DO: Move the front panel to the left side of the screen.
- **36.** Explain: To better facilitate troubleshooting event-based code, we've introduced the event inspector window. This window allows us to view what events have been processed by our event structure, and also see any that are waiting to be processed. If you are new to event programming, you may be interested to know there is a queue working behind the scenes of an event structure to ensure we do not lose any user interaction we'd like to process.
- 37. DO: Move the block diagram to the right side of the screen. Snap to the right side if using Windows 7. Right-click on the event structure border in the "User Interface Loop" and click "Event Inspector Window". Move the Event Inspector Window to the right so front panel and inspector window are side-by-side.



- **38.** Explain: Notice that as I interact with my application, I see those events being captured in real-time. This can help me better understand how my application is behaving. And, the Event Inspector Window allows me to save my event logs to a text file, for documentation purposes.
- **39.** DO: Disable "Log Timeout Events".

Active VIs with	Event Structu	res			
		Mair	n.vi [Whats New	in LabVIEW 2013.	vproj]
Unhandled Eve	ents in Event Q	ueue(s)			
Sequence #	Queue ID	Time	Reference	Event Type	1
26 1 47331963 27 1 47332063 28 1 47332163 29 1 47332263 30 1 47332263 31 1 47332463 32 1 47332663 33 1 47332663 34 1 47332763 35 1 47332863 36 1 47332963	0000000 Tim 0000000 Tim 0000000 Tim 0000000 Tim 0000000 Tim 0000000 Tim 0000000 Tim 0000000 Tim 0000000 Tim 0000000 Tim	eout - eout -			
		ieout -			
37 1 47333063 38 1 47333163	00000000 Tim	eout -			
37 1 47333063 38 1 47333163	00000000 Tim	eout -	[] She	ow Active VIs in <v< td=""><td>i.lib></td></v<>	i.lib>
37 1 47333063 38 1 47333163	t Events	eout -	E Sho	ow Active VIs in <v< td=""><td>i.lib></td></v<>	i.lib>

40. DO: Increase the "Application Timing (ms)" to 2000ms or greater. This will slow down the event handling loop to better show events stacking in the event queue.



LABVIEW 2013 NEW FEATURE DEMO SCRIPT, Page 13 of 21

41. DO: Click "Acquire" or scroll the mouse wheel on the graph several times to build up a set of events in the event queue, shown in the Event Inspector Window.

		ICS Malar	TRAIL - N. N.			
		Ividin.v	/i [whats ivew	in Labview 2015.ivproj		
Jnhandled Ev	ents in Event Q)ueue(s)				
Sequence #	Queue ID	Time	Reference	Event Type	Source Name	Othe
1786	6	47771878	15700000	User Event (high priori	Alarm User?	True
1771	1	47769056	05c00001	Value Change	"Acquire Temperatu	
1772	1	47769243	05c00001	Value Change	"Acquire Temperatu	
1773	1	47769452	05c00001	Value Change	"Acquire Temperatu	
1774	1	47770142	05e00003	Mouse Wheel	"Temp Graph"	
1775	1	47770208	05e00003	Mouse Wheel	"Temp Graph"	
1776	1	47770258	05e00003	Mouse Wheel	"Temp Graph"	
1777	1	47770323	05e00003	Mouse Wheel	"Temp Graph"	
1778	1	47770366	05e00003	Mouse Wheel	"Temp Graph"	
1779	1	47770417	05e00003	Mouse Wheel	"Temp Graph"	
1780	1	47770567	05e00003	Mouse Wheel	"Temp Graph"	
1781	1	47770615	05e00003	Mouse Wheel	"Temp Graph"	
1782	1	47770650	05e00003	Mouse Wheel	"Temp Graph"	
26 1 4733196 27 1 4733206 28 1 4733216 29 1 4733226	3 00000000 Tim 3 00000000 Tim 3 00000000 Tim 3 00000000 Tim	neout - neout - neout - neout -				
26 1 4733196 27 1 4733206 28 1 4733216 29 1 4733226 30 1 4733226 31 1 4733236 31 1 4733246	3 0000000 Tim 3 0000000 Tim 3 0000000 Tim 3 0000000 Tim 3 0000000 Tim 3 0000000 Tim	neout - neout - neout - neout - neout -				
26 1 4733196 27 1 4733206 28 1 4733216 29 1 4733226 30 1 4733236 31 1 4733246 32 1 4733256 33 1 4733266	3 0000000 Tim 3 0000000 Tim	neout - neout - neout - neout - neout - neout - neout -				
26 1 4733196: 27 1 4733206: 28 1 4733216: 29 1 4733226: 30 1 4733236: 31 1 4733246: 32 1 4733256: 33 1 4733266: 33 1 4733266: 34 1 4733276:	3 0000000 Tim 3 0000000 Tim	heout - heout - heout - heout - heout - heout - heout - heout - heout - heout -				
26 1 4733196; 27 1 4733206; 28 1 4733216; 29 1 4733226; 30 1 4733236; 31 1 4733246; 32 1 4733256; 33 1 4733266; 33 1 4733266; 34 1 4733276; 35 1 4733286;	3 0000000 Tim 3 0000000 Tim	neout - neout - neout - neout - neout - neout - neout - neout - neout - neout -				
26 1 47331963 27 1 47332063 28 1 47332163 29 1 47332263 30 1 47332363 31 1 47332463 32 1 47332663 33 1 47332663 33 1 47332663 34 1 47332763 35 1 47332863 36 1 47332963	3 0000000 Tim 3 0000000 Tim	neout - neout -				
26 1 47331963 27 1 47332063 28 1 47332163 29 1 47332263 30 1 47332363 31 1 47332463 32 1 47332663 33 1 47332663 33 1 47332663 34 1 47332663 35 1 47332663 35 1 47332663 36 1 47332963 36 1 47332963 37 1 47333063	3 0000000 Tim 3 0000000 Tim	Neout - Heout -				
26 1 4733196; 27 1 4733206; 28 1 4733216; 29 1 4733236; 30 1 4733236; 31 1 4733246; 32 1 4733246; 33 1 4733266; 33 1 4733266; 35 1 4733266; 35 1 4733266; 36 1 4733296; 37 1 4733306; 38 1 4733316;	3 0000000 Tim 3 0000000 Tim	neout - neout -				
26 1 47331963 27 1 47332063 28 1 47332163 29 1 47332263 30 1 47332263 30 1 47332663 31 1 47332663 33 1 47332663 33 1 47332663 35 1 47332863 36 1 47332863 37 1 47333063 37 1 47333163 18 1 47333163	3 0000000 Tim 3 0000000 Tim	Leout - Heout - Heo	'Acquire Tem	perature Data"		
26 1 4733196 27 1 4733206 28 1 4733216 20 1 4733226 30 1 4733226 30 1 4733226 31 1 4733226 33 1 473326 33 1 4733266 34 1 4733266 35 1 4733266 36 1 473326 38 1 4733316 37 1 4733316 37 1 4733316 37 1 477381 4 47788 4 1 47788 4 1 4768 1 47768	3 0000000 Tim 3 00000000 Tim 3 0000000 Tim 3 0000000 Tim 3 0000000 Tim 3 0000000 Tim 3 0000000 Tim	reout - reout	'Acquire Temy	perature Data" rm User? True		
26 1 4733196. 27 1 4733206. 28 1 4733216. 29 1 4733226. 30 1 4733226. 31 1 4733246. 32 1 4733246. 33 1 4733266. 34 1 4733276. 35 1 4733266. 36 1 4733296. 37 1 4733306. 38 1 4733316. 1768 1 47768. 1769 1 47768.	3 0000000 Tim 3 0000000 Tim 3 00000000 Tim 3 0000000 Tim 3 000000 Tim 3 0000000 Tim 3 0000000 Tim 3 0000000 Tim 3 0000000 Tim 3 0000000 Tim 3 0000000 Tim 3 000000 Tim 3 0000000 Tim 3 000000 Tim 3 000000 Tim 3 000000 Tim 3 000000 Tim 3 0000000 Tim 3 000000 Tim 3 0000000 Tim 3 00000	reout - reout - laue Change "	'Acquire Temj n priority) Alaı 'Acquire Temj	perature Data" rm User? True perature Data"		
26 1 4733196: 27 1 4733206:28 1 4733206: 28 1 4733206: 30 1 4733236: 30 1 4733236: 31 1 4733236: 33 1 473326: 33 1 473326: 33 1 473326: 34 1 473326: 35 1 473326: 37 1 473326: 37 1 473326: 37 1 4733306: 38 1 473336: 1769 1 47768: 1769 1 47768: 1769 1 47768: 1769 1 47768:	3 0000000 Tim 3 0000000 Tim 3 0000000 Tim 3 00000000 Tim 3 0000000 Tim 3 00000000 Tim 3 0000000 Tim 3 00000000 Tim 4 0000000 Tim 4 000000 Tim 4 000000 Tim 4 0000000 Tim 4 000000 Tim 4 000000 Tim 4 0000000 Tim 4 00000000 Tim 4 00000000 Tim 4 00000000 Tim 4 000000000000000000000000000000000000	reout - reout -	'Acquire Tem n priority) Alaı 'Acquire Tem Shc	perature Data" rm User? True perature Data" ow Active VIs in <vi.lib></vi.lib>		

- **42. Explain:** If anyone has done event-based programming in past, they may be familiar with user events. These are simply events which are generated by logic in our program, rather than by a UI interaction. If anyone has taken the new Core 3 training, which uses the Queued Message Handler architecture, this leverages user events.
- **43.** Explain: To build on templates and sample projects, and to encourage best practice architecture choices using event-based programming, we've implemented a highly requested feature, which is the ability to specify the priority of a user event. I mentioned there is a queue behind the scenes of an event structure, ensuring we do not lost any UI interaction from one loop iteration to the next. In past, user events always received the same priority as other events. This mean that system-critical tasks, such as a shutdown to avoid overheating, might require an extra queue or notifier in a large application, so they could be given the correct response priority.
- **44. Explain:** In LabVIEW 2013, we're helping users leverage the event-based architectures they are implementing, and handle these system-critical tasks with less development effort.
- **45. DO:** Show the block diagram, and make sure the "Acquire" case is shown in the "Processing Loop"

- **46. Explain:** In this example application, I'm going to alarm my user if the result of my temperature data acquisition and subsequent analysis is above the limit they've specified. However, I'm going to consider this a system-critical response, something that must be processed before all other actions, in the event an operator shutdown was required in response to this excessive temperature. I'm facilitating this with a high priority user event.
- **47. DO:** Point out the temperature comparison and the "Generate User Event" function, which generates a "high priority" user event when called.



- **48. Explain:** To show the way LabVIEW responds to high priority events, I've slowed down this application so we can observe what happens in the event queue when this high-priority event occurs.
- **49. DO:** Bring the Event Inspector Window to the front so the UI and Event Inspector are sideby-side. Clear all events and ensure that timeouts are not being logged. Click the "Acquire" button, and then initiate several mouse wheel events on the "Temperature Graph" to fill up the event structure queue.
- **50. Explain:** Notice that when a high priority event is fired, it automatically goes to the top of the event structure's list of items to process. Thus, we can make efficient use of this already-in-

		Main.v	/i [Whats New	in LabVIEW 2013.lvproj]	
Jnhandled Eve	ents in Event Q	ueue(s)	Reference	Event Tune	Source Name	Oth
Sequence #	Queue ID	10054070	1.5700000	Event Type	Source Name	- Un
2054	0	48054878	15/00000	User Event (high priori	Alarm User?	True
2049	1	48054719	05e00003 -	Mouse wheel	Temp Graph	
2050	1	48054762	05e00003	Mouse Wheel	"Temp Graph"	
2051	1	48054808	05e00003	Mouse Wheel	"Temp Graph"	
2052	1	48054847	05e00003	Mouse Wheel	"Temp Graph"	
2053	1	48054875	05e00003	Mouse Wheel	"Temp Graph"	
2055	1	48054907	05e00003	Mouse Wheel	"Temp Graph"	
2056	1	48054932	05e00003	Mouse Wheel	"Temp Graph"	
2057	1	48054968	05e00003	Mouse Wheel	"Temp Graph"	
vent Log	80.0EF00003 N	Anuse Wheel "	Tama Granh"			

place structure to handle system-critical responses.

- 51. DO: Close the block diagram
- **52. Explain:** We talked about some new development environment enhancements, available in LabVIEW 2013. I'd like to shift over toward the subject of deployment.
- 53. Explain: For anyone needing secure remote access to LabVIEW applications, whether running on a desktop or embedded target, web services are a great option for this communication. They allow SSL security, are industry standard and may be accessed by thin clients over a network, and are IT friendly. Web Services have been available in LabVIEW since 8.6, but we've gone to great lengths to simplify the process of creating, deploying, and debugging web services.
- **54. Explain:** Web Services are now created as a project item. To do this, we simply right-click on "My Computer", then choose "New" and "Web Service" which creates a new item for us.
- 55. DO: Steps listed above

🔁 Whats New in La	bVIEW 2013.lvproj * - Proje	ct Explo				
File Edit View	Project Operate Tools	Window Help				
*h 🗃 🞒 X	🗅 🗈 X 🛛 🖬 🖌	🖽 🕶 🚰 🔥 📗 😹 🍅 🍝				
Items Filer	Items Files					
🖃 📴 Project: W	hats New in LabVIEW 2013.lv	rproj				
	New 🕨	VI				
🕀 📁 Sup	Export •	Virtual Folder				
🗄 🙀 Bro	Import •	Control				
Ma		Library				
🖽 😤 Dep	Add 🕨	Variable				
	Find Project Items	Class				
	Arrange By	XControl				
	Expand All	Web Service				
	Collapse All	NI-DAQmx Task				
	Help	NI-DAQmx Channel				
	Properties	NI-DAQmx Scale				
		Targets and Devices				
		New				

- 56. Explain: I've already added a web service, to show the new experience.
- 57. DO: Expand "Broadcast Alarm" web service project item and "Web Resources" folder.



58. DO: Open the "Web Service VI.vi", move both front panel and block diagram to the right side of the screen and overlay as shown.



- **59.** Explain: In LabVIEW 2013, not only may I create and deploy web services to remotely access my LabVIEW applications, but I am able to debug the web method VIs, comprising that web service, directly from the LabVIEW project. This is an all new feature of LabVIEW 2013 web services. Moreover, you'll notice this web service VI uses global variables. LabVIEW 2013 web services are created in the same namespace as regular LabVIEW applications, and therefore may use global variables and functional global variables to communicate, where before we'd have need to use TCP/UDP etc. to facilitate this cross-environment communication.
- 60. Explain: I can "publish" my web service, like you might be used to in LabVIEW.
- **61.** DO: Right-click the "Broadcast Alarm" web service name, choose "Application Web Server" and show the options to "publish" and "unpublish".



62. Explain: But, in LabVIEW 2013, I can begin a debugging session on my web method VI. I do this by right-clicking my "Broadcast Alarm" web service project items and choosing "start". Notice that my Web method VI is now reserved and waiting to run.

63. DO: Right-click "Broadcast Alarm" and choose "start". Then, point out the VI being reserved.



- **64.** Explain: I may now use standard LabVIEW debugging tools to understand the behavior of my web service. To run the web service, I may obtain the URL of my web service by rightclicking on my web method VI.
- **65.** DO: Turn on highlight execution for the Web Service VI. Right-click on Web Service VI.vi (GET) and choose "Show Method URL". Mention that LabVIEW automatically generates the URL for you. [This used to be a pain point for customers] Click on "Copy URL" and <u>close the "HTTP Method URL" dialog</u>.

Wy Computer Subyoting Documents Web Resources Web Service VLvi (GET	Open		
- J Startup VIs	Explore	HTTP Method URL	
White WebService1	Show in Files View Ctrl+E		
🖶 😨 Dependencies	Brint	Available Servers	
🗄 📸 Build Specifications	Phnt	8001 - Local Debugging	
	Run	HTTP Method URL	
	Find 🕨	http://127.0.0.1:8001/Broadcast_Alarm/Web_Service_VI	
	Save		
	Save As		ConvillBl
	Show Method URL		Close Close

- **66.** DO: Open a browser and snap it to the left-side of the screen so both the browser window and Web Service VI block diagram are shown at a time. Paste the URL into the browser's URL bar and hit enter.
- **67. Explain:** This starts my web method, allows me to observe the web service VI's behavior with standard LabVIEW debugging tools [in this case highlight execution], and ultimately I

get my results in the browser window.



- **68.** DO: Close the browser, close the web service VI, right-click on "Broadcast Alarm" web service item and choose "stop" to end debug session.
- **69.** DO: Expand the "Whats New in LabVIEW 2013" project's build specifications, right-click on "LabVIEW 2013 EXE" and choose "Properties".
- 70. Explain: We did not forget about deployment when it comes to web services. In past there were many preparations necessary to deploy an application (EXE) which leveraged a web service. In 2013, we've streamlined this process by including a web service category in EXE build specs. Now, you can choose any web service included in your project, select SSL options, and build this web service into your EXE. It will then deploy automatically when the EXE runs, making remote access to both desktop and embedded LabVIEW applications a

Category	Web Services
Information Source Files Destinations Source File Settings Icon Advanced Additional Exclusions Version Information Windows Security Shared Variable Deployment Run-Time Languages Web Services Pre/Post Build Actions Preview	Web service name Broadcast Alarm Broadcast Alarm Enable HTTP Enable HTTP Enable SSL HTTP port SSL port 8002 SSL certificate file Query host for certificates Server address Query

- 71. DO: Close the EXE build.
- **72. Explain:** Now I'm going to switch back to slides and show you a few more deployment related features we've added in LabVIEW 2013.