NOTE: This is a Beta build of LabVIEW NXG 2.0. You may experience unexpected lag or crashes. Please notify an NI Employee if you have questions, comments or feedback.

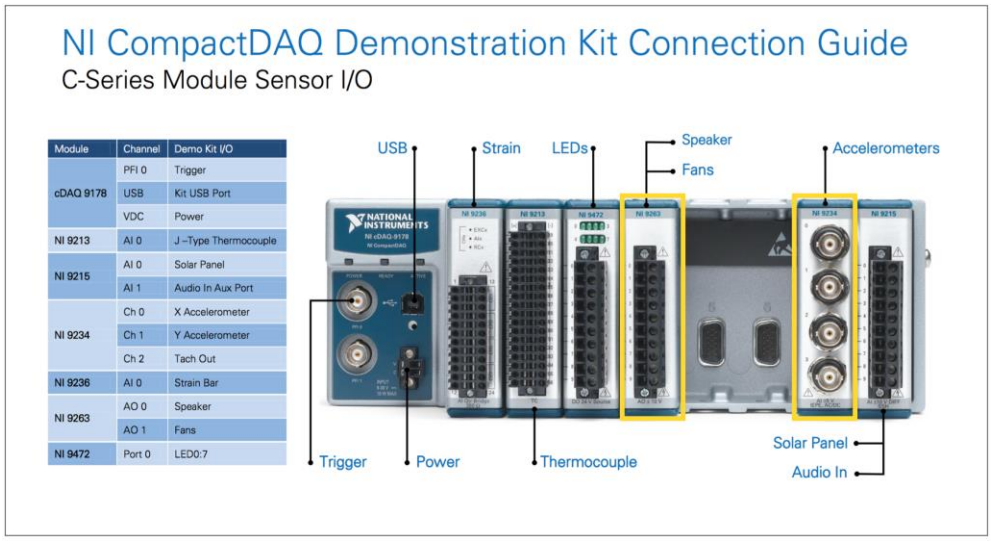
LabVIEW™ NXG

LabVIEW NXG 2.0 Test Drive

NOTE: This is a Beta build of LabVIEW NXG 2.0. You may experience unexpected lag or crashes. Please notify an NI Employee if you have questions, comments or feedback.

LabVIEW NXG 2.0 Test Drive

Hardware



Module	Channel	Demo Kit I/O
cDAQ 9178	PFI 0	Trigger
	USB	Kit USB Port
	VDC	Power
NI 9213	AI 0	J –Type Thermocouple
NI 9215	AI 0	Solar Panel
	AI 1	Audio In Aux Port
NI 9234	Ch 0	X Accelerometer
	Ch 1	Y Accelerometer
	Ch 2	Tach Out
NI 9236	AI 0	Strain Bar
NI 9263	AO 0	Speaker
	AO 1	Fans
NI 9472	Port 0	LED0:7

- CompactDAQ 9178 chassis – CompactDAQ is a portable, rugged DAQ platform that integrates connectivity and signal conditioning into modular I/O for directly interfacing to any sensor or signal.
    - Modules we will be using today:
        - 9234 – 4-channel acceleration input to measure vibration
        - 9263 – 4-channel analog voltage output to control fan speed

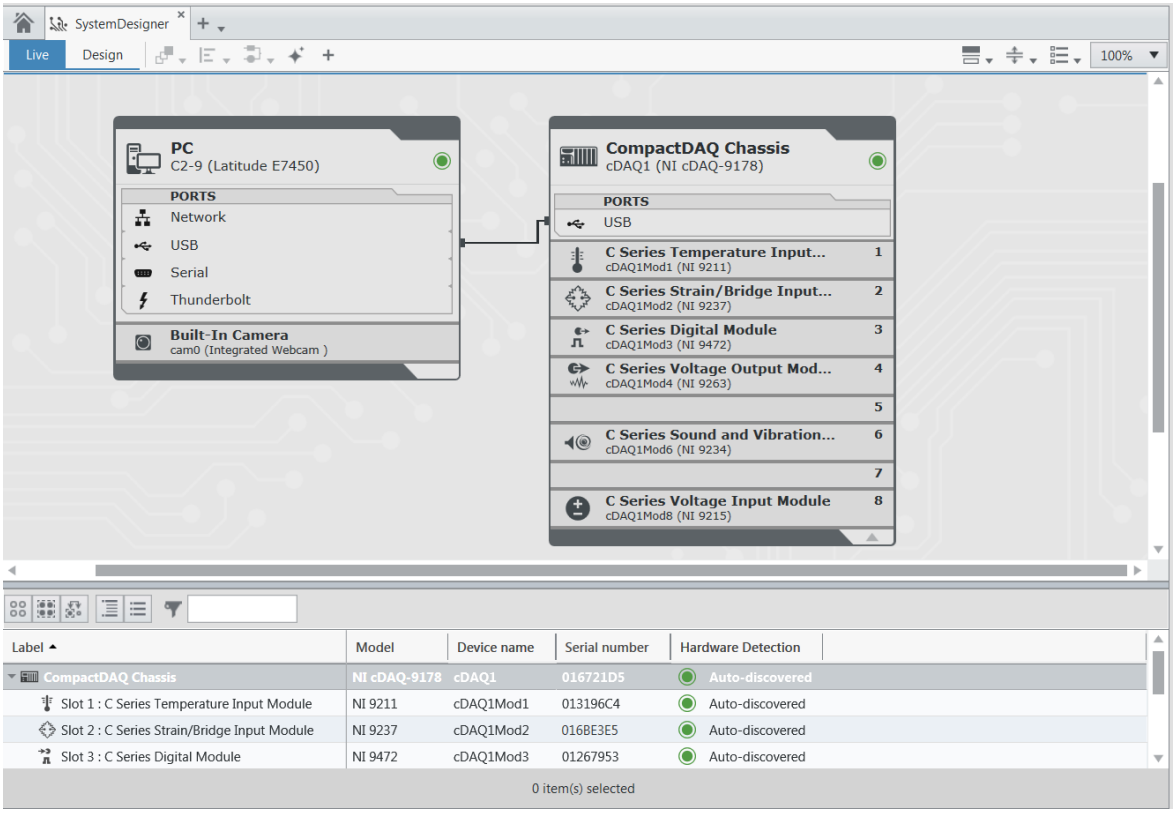
- Sound and Vibration Signal Simulator – this is a demonstration board which generates sound and vibration signals. There are two fans on the board; one is balanced, while the other is unbalanced. You can change which fan is active with a switch on the board. Finally, you can change the speed of the fan with a knob or with a constant analog voltage between 0 and 10 volts. Change the control scheme with a switch on the board.

NOTE: This is a Beta build of LabVIEW NXG 2.0. You may experience unexpected lag or crashes. Please notify an NI Employee if you have questions, comments or feedback.

Activity 1

Complete the following steps to build a simple application to control and measure the speed of a fan. These exercises are just a guide, explore the LabVIEW NXG 2.0 software as you go. Ask an NI Employee if there's something you can't find in NXG and would like to see.

1. **Discover Your Hardware**
   Navigate to the Hardware section of LabVIEW NXG to open the SystemDesigner view and build functional representation of your system.



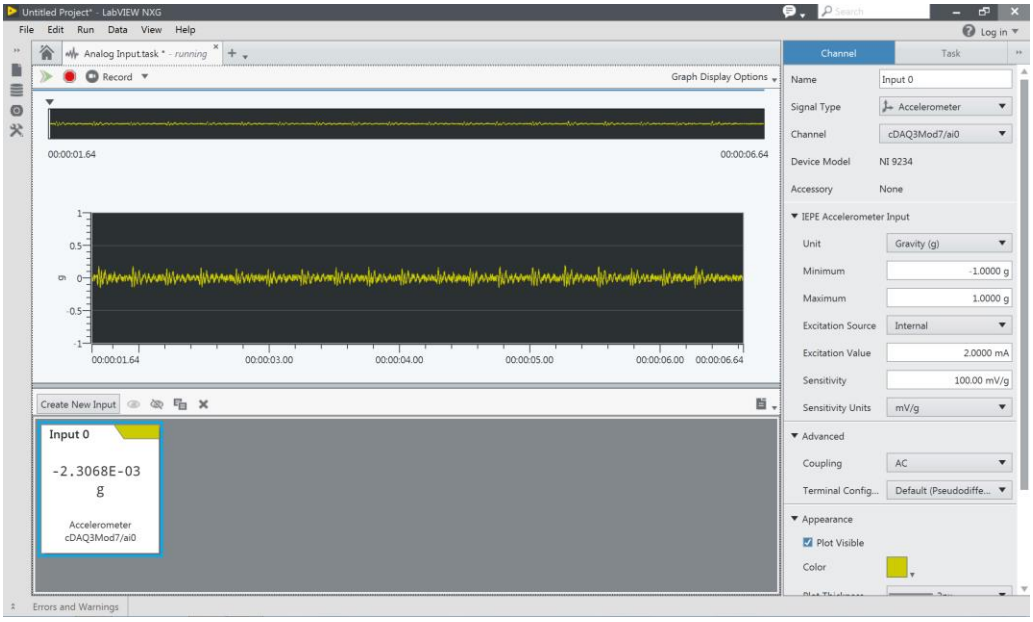
   Tips for exploring the software: investigate the differences between the Live and Design views.

2. **Validate Your Signals**

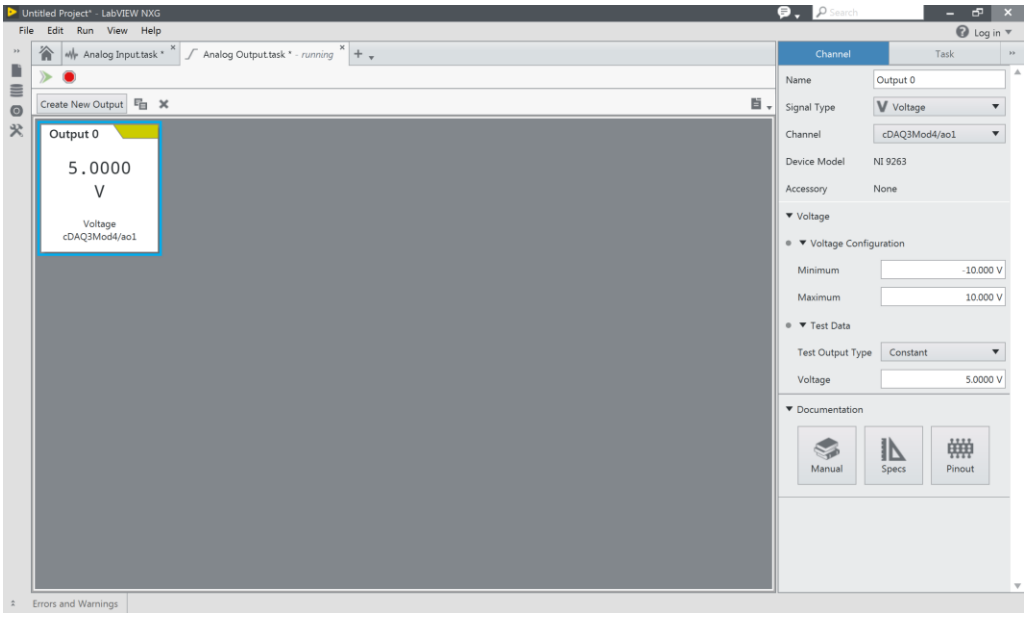
NOTE: This is a Beta build of LabVIEW NXG 2.0. You may experience unexpected lag or crashes. Please notify an NI Employee if you have questions, comments or feedback.

Use two Measurement Panels to connect to your hardware and validate that you can perform the following measurements:

- Receive vibration/acceleration data in units of g



- Output voltage to control the fan



3. Automate Fan Control and Vibration Measurement

Make a VI that continuously receives user input to change fan speed and reads vibration data.

NOTE: This is a Beta build of LabVIEW NXG 2.0. You may experience unexpected lag or crashes. Please notify an NI Employee if you have questions, comments or feedback.

Save the VI as 'Fan_Speed.gvi'

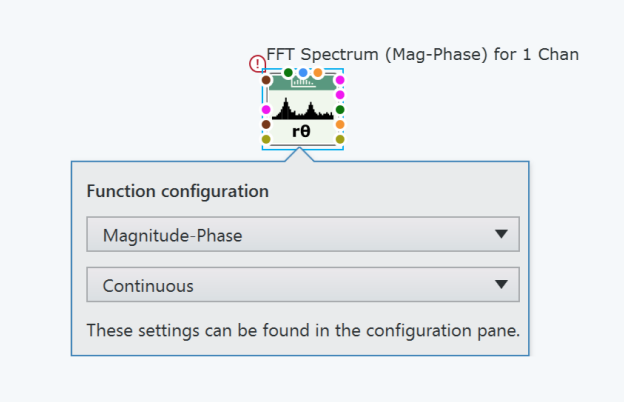
You can reuse the hardware configuration for the Analog Input and Output tasks in your VI by dragging and dropping the tasks on to the diagram.

If you would like to explore the use of libraries in LabVIEW NXG you can add your VI to a library.

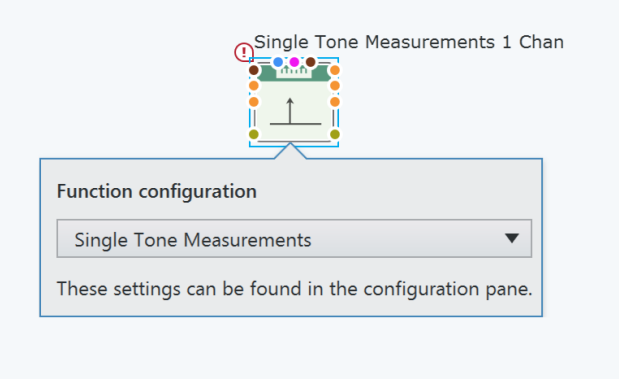
4. Interpret Vibration Data (Optional)

Modify the application to analyze vibration frequencies, allowing the user to see which frequency is the most prominent at different fan speeds. Determine the vibration frequency with the highest amplitude and compare it against a user-defined limit. Display pass/fail to the user.

- Plot a graph that shows the magnitude of vibration frequencies using the FFT Spectrum VI. The graph should not show frequencies above 1kHz.



- Use the Tone Measurements VI to find the vibration frequency with the highest amplitude.

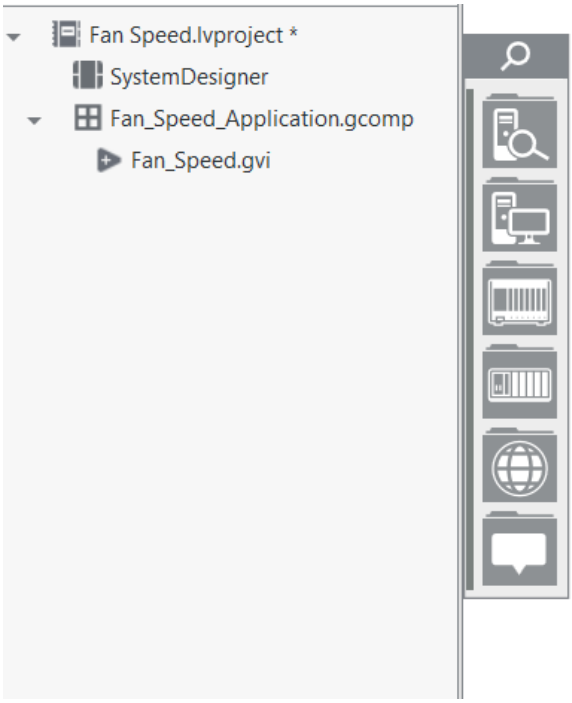


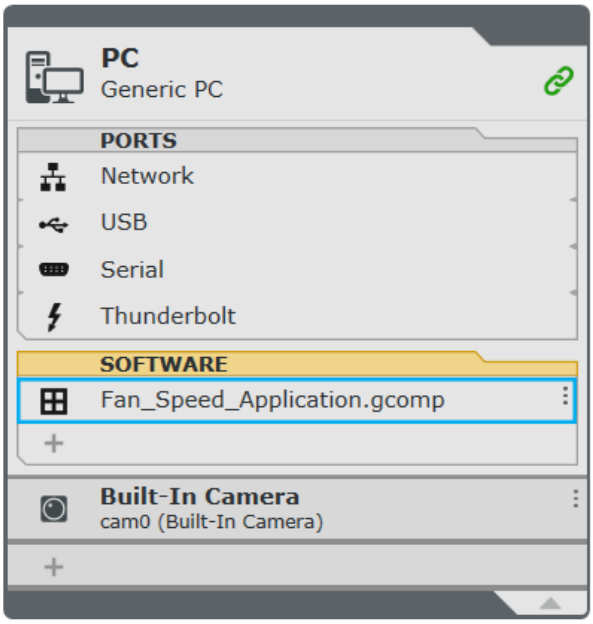
5. Create an Executable with Application Builder

NOTE: This is a Beta build of LabVIEW NXG 2.0. You may experience unexpected lag or crashes. Please notify an NI Employee if you have questions, comments or feedback.

Application builder can be used to create and executable from SystemDesigner. You can't build a single VI into an application, the VI must belong to either a Library or an Application which you can add to your project from the Navigation Pane or through the Design view of SystemDesigner.

Note: VIs must be saved before you can add them to a Library or Application



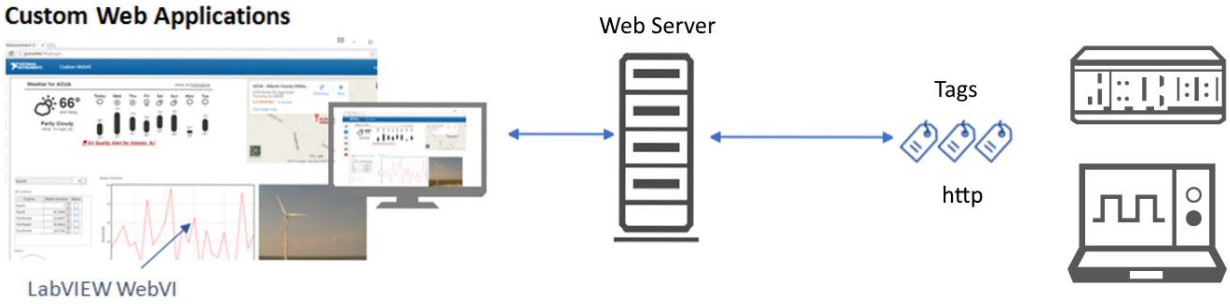


NOTE: This is a Beta build of LabVIEW NXG 2.0. You may experience unexpected lag or crashes. Please notify an NI Employee if you have questions, comments or feedback.

Activity 2 – WebVIs

This exercises will be a brief introduction to WebVIs. We'll look at how we can make a simple WebVI for an existing, LabVIEW 2017 code.

This is an overview of the structure of this application. The Web Server can by any Web Server such as the LabVIEW 2017 Web Server, LabVIEW NXG Web Server, Amazon Web Server. In this example, we are using the LabVIEW 2017 Web Server.

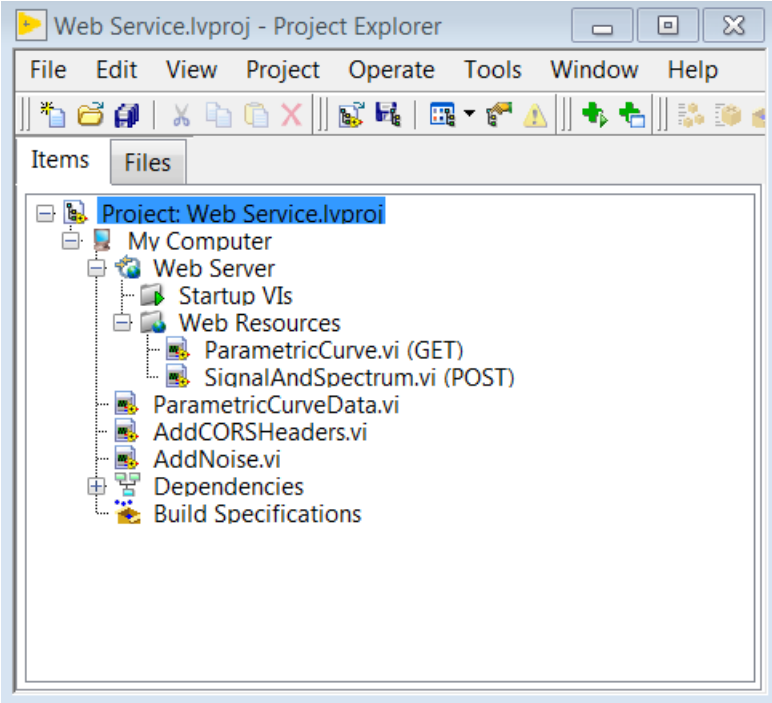


1. **Open the pre-built LabVIEW 2017 project**
   Open Web Service.lvproj from C:\Seminars\LabVIEW NXG Test Drive\WebVIs\Web Service

   This application contains a LabVIEW 2017 web server that has several functions we can utilise in our LabVIEW NXG WebVI. This LabVIEW 2017 code could be anything, including a control system for hardware that you want to control remotely.

   The LabVIEW 2017 Web Server contains some code that publishes data so it can be accessed remotely. It also can accept data from external sources that it can give to the LabVIEW 2017 VIs.

NOTE: This is a Beta build of LabVIEW NXG 2.0. You may experience unexpected lag or crashes. Please notify an NI Employee if you have questions, comments or feedback.



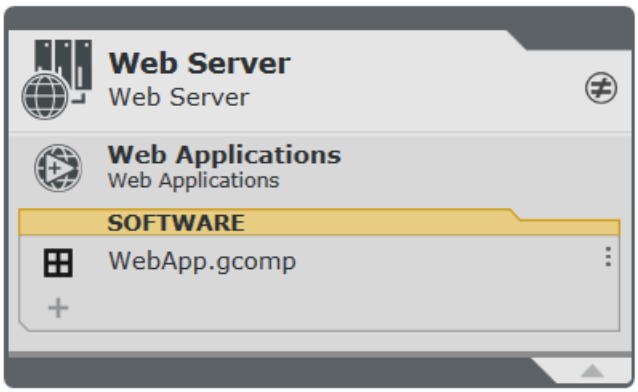
Right click the web server and select start, then OK.

2. Design the WebVI in LabVIEW NXG

This demo code, and other WebVI examples, can be found at https://github.com/ni/webvi-examples

Open Call LabVIEW Web Service.lvproject from C:\Seminars\LabVIEW NXG Test Drive\WebVIs\WebVI

Go to SystemDesigner



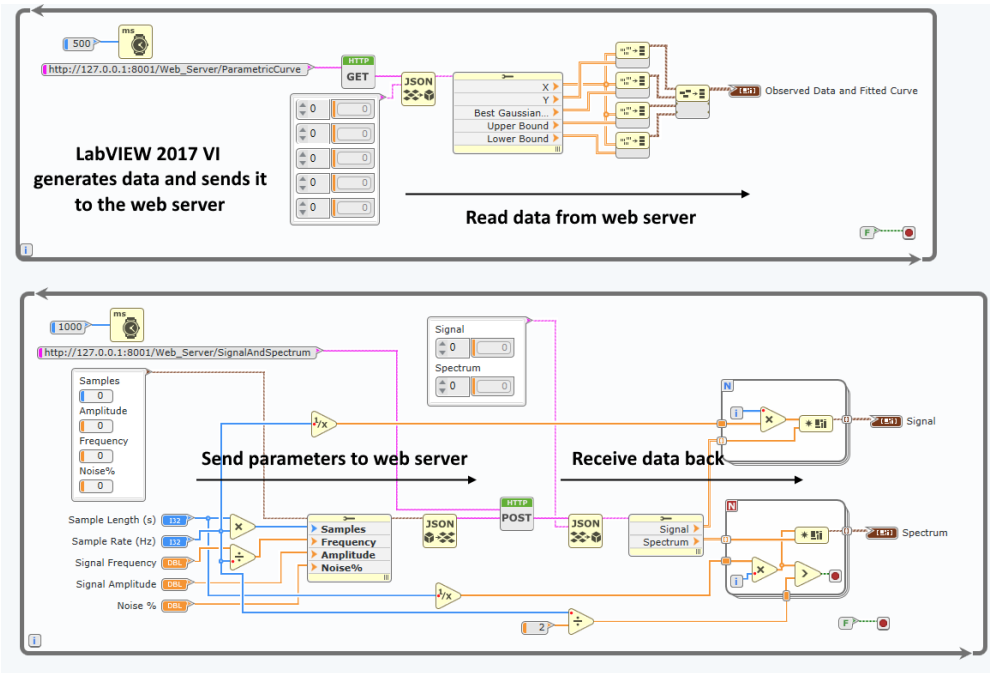
Here we can see the WebVI Application. We have a Web Application that can be built but first we need to finish the WebVi.

NOTE: This is a Beta build of LabVIEW NXG 2.0. You may experience unexpected lag or crashes. Please notify an NI Employee if you have questions, comments or feedback.

Open the WebApp.gcomp and then open Main.gviweb.

Switch to the Diagram tab

This VI is sending data from the Front Panel to the LabVIEW 2017 Web Server using HTTP functions. It also reads data, that the LabVIEW 2017 application has created, from the Web Server to update the graphs.



Switch to the Panel tab

There are two unplaced Front Panel controls in the Unplaced Items Tray. Add the Signal Frequency and Noise % controls to the front panel. Notice how they look different to the front panel items you used in activity 1. That's because they are HTML controls that can work in a web browser.

Switch to the HTML tab

This is the HTML that is created as you build your Front Panel. And we can make edits directly to this code.

NOTE: This is a Beta build of LabVIEW NXG 2.0. You may experience unexpected lag or crashes. Please notify an NI Employee if you have questions, comments or feedback.

Scroll down to find the space between </head> and <body>. Add the following HTML code between </head> and <body> to apply a CSS (cascading style sheet) file to change the front panel appearance.

<link href="keynote-styles.css" rel="stylesheet" type="text/css">

```
        /* End read only section. To add additional styles, add a new style tag after this one. */
    </style>

</head>

<link href="UI-style.css" rel="stylesheet" type="text/css">

<body>
    <div id="ni-outdated-browser-message" style="display:none">This browser is reporting compatibility
    <div id="ni-failed-to-load-vireo-source" style="display:none">
        <div id="ni-failed-to-load-vireo-message-title">
            Unable to load resource file
        </div>
        <div id="ni-failed-to-load-vireo-message-body">
            Verify all WebVI files are available and served using an HTTP server or use a browser that
        </div>
```

Click Apply

Run the VI and confirm the application is working.

3. **Build the WebVI for deployment**

Now we can build the web application. Go to SystemDesigner and right click the WebApp.gcomp and select Build

When the bottom, Build Queue, pane shows the build is complete, right click WebApp_Web Server and select Locate item in Windows Explorer.

Open Main.html in FireFox

Activity 3 – Code Conversion Utility

1. **Create some LabVIEW 2017 code to convert**

Open any example project or example VI from the LabVIEW 2017 template projects or examples.

Save the project to the desktop

2. **Convert the project**

NOTE: This is a Beta build of LabVIEW NXG 2.0. You may experience unexpected lag or crashes. Please notify an NI Employee if you have questions, comments or feedback.

Convert the project/file using the Code Conversion Utility by going to Projects > Convert a Project on the LabVIEW NXG 2.0 homescreen.

Select your files.

Convert the project. Explore the conversion report, what worked, what didn't.