Exercise 6-3: Create an Installer

Goal

Create an installer build specification and build the installer. As a challenge, remotely debug the application created by the installer.

Hardware Setup

(Hardware) In the exercises where we work with Analog Input/Output channels, we use PCI-6221/USB-6212 multifunction I/O device paired with the BNC-2120 shielded connector block. Analog Input 2 should be connected to the Sine/Triangle BNC connector. Analog Input 3 should be connected to the TTL Square Wave BNC connector. The Sine/Triangle waveform switch should be set to Sine.

Scenario

Creating an installer simplifies deploying an application to multiple machines. After you have prepared your files, you create an Application (.exe) Build Specification and then create an Installer Build Specification.

Design

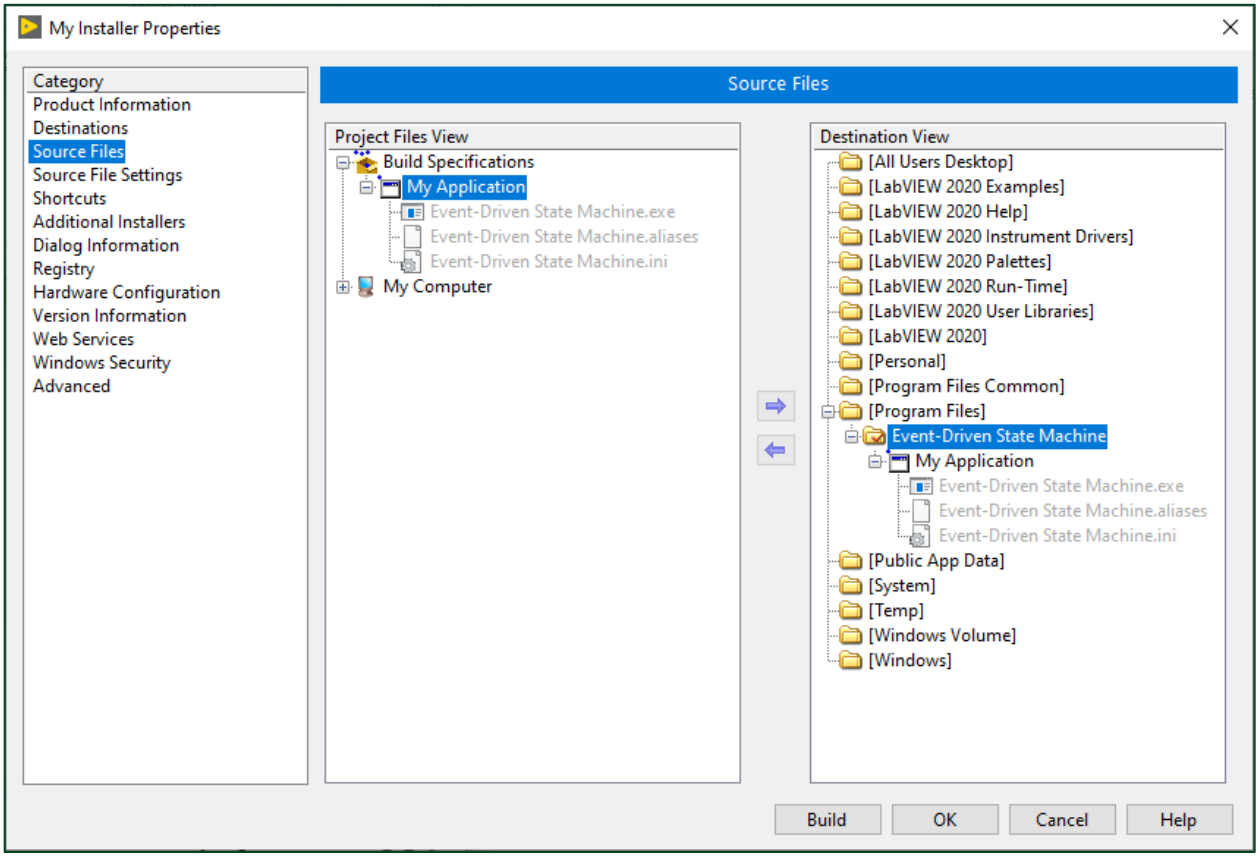
Use an Installer Build Specification to create an installer for the Application (.exe) Build Specification.

Warning: To successfully build an installer, make sure that your exercise source files are placed in the appropriate location, that is — C:\Exercises\ LabVIEW Core 2\"name of the current exercise".

Guided Instructions
Create an Installer Build Specification

1. Open C:\Exercises\LabVIEW Core 2\Creating an Installer\Event-Driven State Machine.lvproj.
2. Right-click **Build Specifications** in the **Project Explorer** window and select **New» Installer** from the shortcut menu.
3. Modify the installer destination in the **Product Information** page.
    - Select the **Product Information page**.
    - Type C:\Exercises\LabVIEW Core 2\Creating an Installer\Installer as the Installer destination.
4. Specify the Installer Build Specification.
    - Click the **Source Files** page.
    - Select **My Application** under the Build Specification folder.
    - Expand **Program Files** and select **Event-Driven State Machine** in the **Destination View** tree.

- Click the right arrow next to the **Project Files View** tree to place the event-driven state machine application and all application support files under **Program Files» Event-Driven State Machine** in the **Destination View** tree as shown in figure



below.

5. Add the NI LabVIEW Run-Time Engine to the installer by modifying the **Additional Installers** page.
   - Select the **Additional Installers page**.
   - Select the **NI LabVIEW RunTime 2020** and **NI-DAQmx Runtime 20.0 installers** (or later versions), if they are not already automatically selected.
   - Click **OK**.

   Note: Some additional installers you might select in the **Additional Installers** page, can require you to have downloaded the product installation package before you can include the product installer in your application installer. To download an installation package, go to ni.com/info and enter the code downloads.

6. In the **Project Explorer** window, right-click the **My Installer** build specification and select **Build** from the shortcut menu.
7. Click **Done** when LabVIEW finishes building the installer.
8. Save and close the **Project Explorer** window and close LabVIEW.

Test

1. Run the install.exe file in the C:\Exercises\LabVIEW Core 2\Creating an Installer\Installer\Volume directory.
2. Follow the instructions on-screen to install the application. By default, the application is created inside the C:\Program Files (x86)\Event-Driven State Machine directory.
3. From the Start menu, navigate to **Start» Event-Driven State Machine** or from the Windows Explorer, navigate to C:\Program Files (x86)\Event-Driven State Machine directory.
4. Right-click **Event-Driven State Machine** and select **Run as administrator**.

Note: You must run the application as an administrator because the application creates a file in the <Program Files> folder. If you do not run the program as an administrator, Error 8 occurs.

5. Click **Yes** in the dialog box asking for permission to make changes to the computer.
6. Test the application by clicking the **Acquire** button, **Clean** button, and the **Exit** button.

Challenge (Optional)

If you have Internet access during class, try to debug the application on a remote computer.

1. Verify that classroom has Internet access.
2. Decide whether to debug a classmate's application or install your application on your classmate's computer.
3. If you decide to debug your own application on a remote computer you must distinguish your application from the applications already on your classmate's computer.
    - In the installer build specification, rename your application with a unique name.
    - Transfer your installer to the remote computer using a USB flash drive or the network.
    - Install your application.
4. To use LabVIEW on your computer to debug a running application on a remote computer, you must determine the IP address of the remote computer, also known as the destination computer.

Note: Consider your computer to be the development computer and your classmate's computer to be the destination computer.

    - Open the Windows Start menu on the destination computer.
    - Enter cmd in the search box and press the <Enter> key.
    - Type ipconfig at the prompt in the Command window and press the <Enter> key.
    - Note the IP address.
5. Run the application on the destination computer.
6. On the development computer, launch LabVIEW, if necessary.
7. Select **Operate» Debug Application or Shared Library** from the LabVIEW menu.
8. Enter the IP address of the destination computer in the **Machine name or IP address** text box.
9. Select the application from the **Application or shared library** drop-down menu.
    - Click the **Refresh** button if the application you want does not appear in the list.
10. Click the **Connect** button to create the debugging connection.
11. Start debugging the running application.
    - Open the block diagram.

- Turn on Execution Highlighting.
- Try using probes, breakpoints, and single-stepping.

12. Stop the application by clicking the **Stop** button in the debugging window.

Note: Refer to the *LabVIEW Continuous Integration Tutorial (Jenkins/GitHub)* on ni.com for more details.

End of Exercise 6-3