

Exercise 4-3 Initialize an Application from a File

Goal

Initialize your application by reading configuration data from an INI file.

Scenario

In the sprint planning meeting for this iteration, the product owner chose to implement the following user story:

- As a boiler operator, I want to specify the configuration settings that the boiler controller uses, so that I can reuse the controller for boilers with different configuration settings.

After discussing this user story with the team, you have decided to read the configuration settings from an INI file. The requirements document indicates that the Fuel Control Valve should be limited to values between 10 and 75%. Since this may vary from boiler to boiler, those values should be read from a configuration file instead of written as constants on the block diagram.

Your team identified the following development tasks that must be completed:

- Update `MHL_Data.ct1` to contain the Fuel Control Valve limits that you want to read from the INI file.
- Create the INI file that contains configuration constants for the system.
- Create a VI that allows the user to specify the INI file, reads configuration constants from that file, and passes those values to the rest of the application.
- Modify existing code to use the configuration constants.

Design

While it is possible to create the INI file manually, your team decides that would be a good idea to create a VI that programmatically creates the INI file. That way, you can easily add more constants to the file or reuse the code to create INI files for other boilers that require different constant values.

Develop the INI file and the VI that reads it in a manner that allows for future expansion in case you want to scale it to read other constants for the MHL, the boiler controller, or the boiler. To do this, you create a subVI that reads the boiler constants. If you eventually need to read constants for the other loops, you can create separate subVIs to read the constants for each of those loops.

Implementation

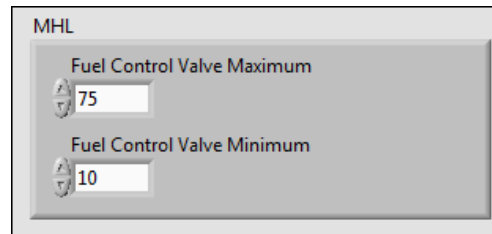
1. Add configuration files to the project.

- ☐ In Windows Explorer, copy the Configuration directory located in the <Exercises>\LabVIEW Core 3\External\Support VI directory and paste it into the <Exercises>\LabVIEW Core 3\Course Project\support directory.
- ☐ Create a virtual folder called Configuration in the Support VIs folder of the Boiler Controller project. Add only Write Configuration Settings File.vi to the virtual folder.

2. Update the INI file with new default values.

- ☐ Open the Write Configuration Settings File VI and verify the **MHL** default values are set to the values shown in Figure 4-15.

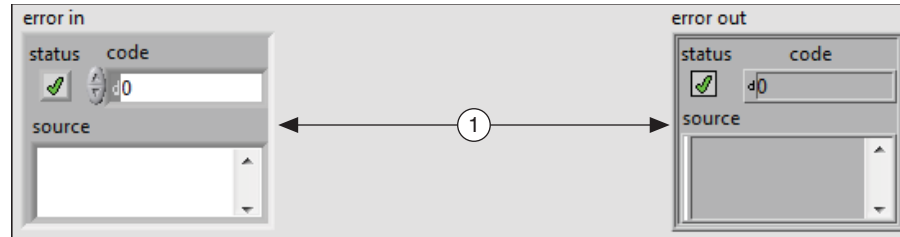
Figure 4-15. Write Configuration Settings File VI Front Panel



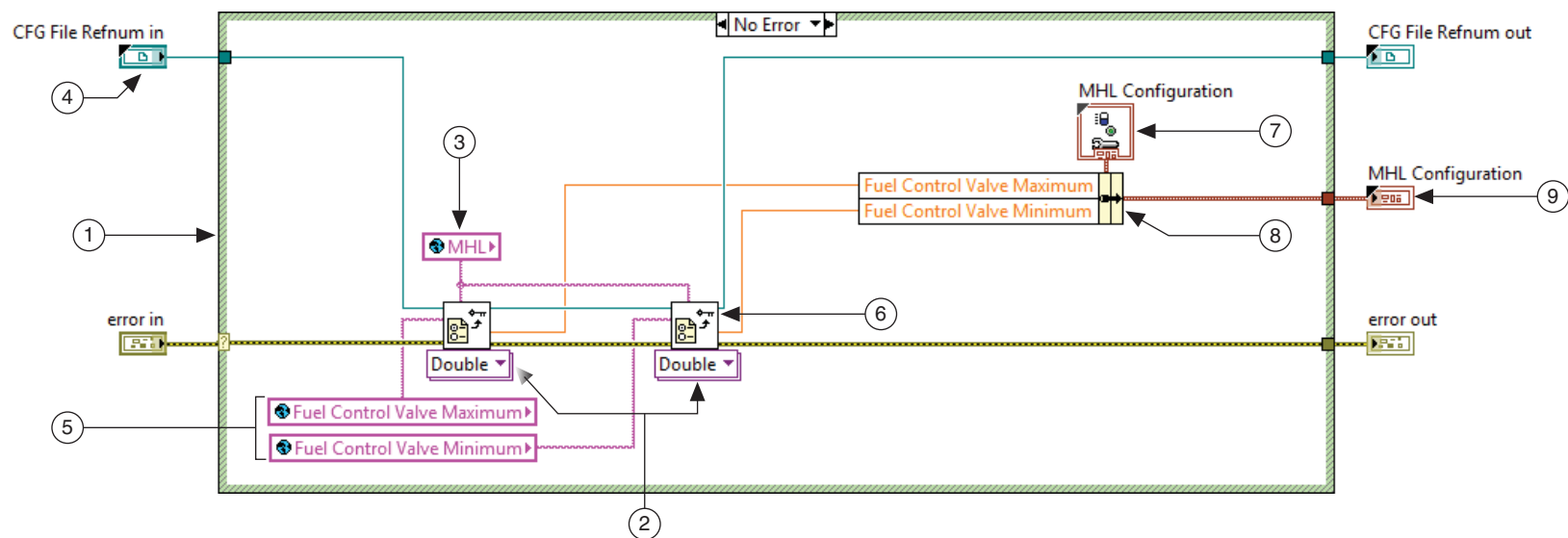
- ☐ Run the Write Configuration Settings File VI and save the file as Boiler Init.ini in the <Exercises>\LabVIEW Core 3\Course Project directory.
- ☐ Add Boiler Init.ini to the project.
 - Right-click **My Computer** in the **Project Explorer** window and select **Add»File** from the shortcut menu.
 - Navigate to Boiler Init.ini and click the **Add File** button to add the file to the project.

3. Create a VI that allows the user to specify the INI file, read configuration constants from that file, and pass those values to the rest of the application.
 - Add a new VI to the project under the **Support VIs»Configuration** virtual folder and save the VI as `Read UI Constants.vi`.
 - Create the front panel and block diagram of the Read UI Constants VI as shown in Figures 4-16 through 4-19.

Figure 4-16. Read UI Constants VI Front Panel

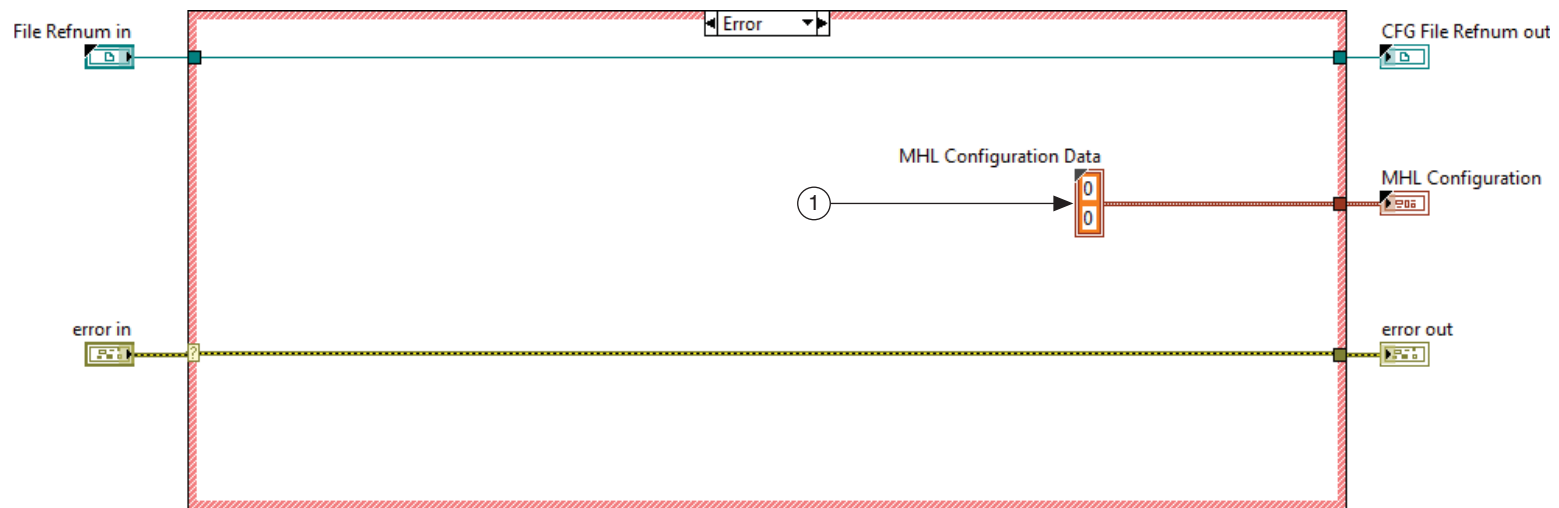


- 1 **Error In and Error Out**—Place an Error In and Error Out control and indicator on the front panel. After you complete the block diagram, the front panel will contain more items.

Figure 4-17. Read UI Constants VI Block Diagram No Error Case

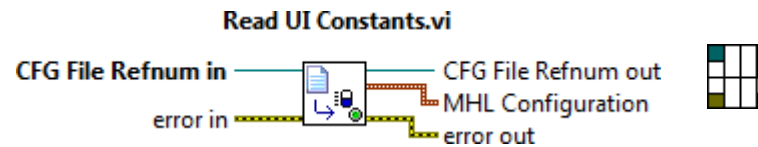
- 1 **Case Structure**—Place a Case structure on the diagram and wire the **error in** cluster to the case selector.
- 2 **Read Key**—This VI reads a value from the key in a MHL section of the configuration data. Select **Double** from the polymorphic selector.
- 3 **Boiler System Globals variable**—Drag **Boiler System Globals.vi** from the **Project Explorer** window and select **MHL** from the list. Right-click the global variable and select **Change To Read** from the shortcut menu. Wire this global variable to the **section** input of each of the Read Key VIs.
- 4 **Refnum**—Right-click the **refnum** input of the Read Key VI and select **Create»Control**. Rename the refnum **CFG File Refnum in** and move the refnum outside the Case structure. Re-wire the refnum if necessary.
- 5 **Boiler System Globals variable**—Change the global variables to read and select **Fuel Control Valve Minimum** and **Fuel Control Valve Maximum** from the list. Wire these variables to the **key** input of the Read Key VI.
- 6 **Refnum**—Right-click the **refnum out** output of the Read Key VI and select **Create»Indicator**. Rename the refnum **CFG File Refnum out** and move the refnum outside the Case structure. Re-wire the refnum if necessary.
- 7 **MHL Configuration constant**—Drag **MHL Configuration.cti** from the **Type Definitions** folder in the **Project Explorer** window.
- 8 **Bundle By Name**—Wire the MHL Configuration constant to the **input cluster** input of the Bundle By Name function. Expand the function to display two terminals. Wire the **value** output of each of the Read Key VIs to the Bundle By Name function.
- 9 **MHL Configuration indicator**—Right-click the **output cluster** output of the Bundle by Name function and select **Create»Indicator**. Rename the indicator **MHL Configuration** and move it outside the Case structure. Wire the Bundle by Name function to the indicator.

Figure 4-18. Read UI Constants VI Block Diagram Error Case



- 1 **MHL Configuration constant**—Right-click the tunnel and select **Create»Constant** to create this constant.

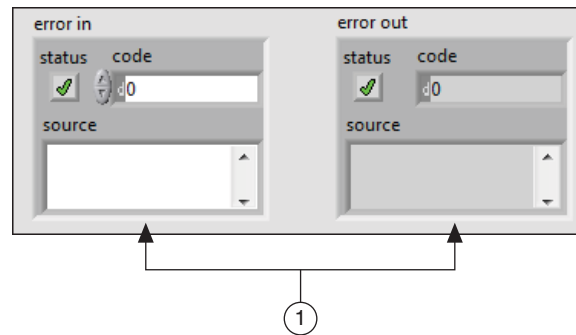
Figure 4-19. Read UI Constants VI Connector Pane



- 1 Filter glyphs by keywords, file, get, and control to find appropriate glyphs for the icon.

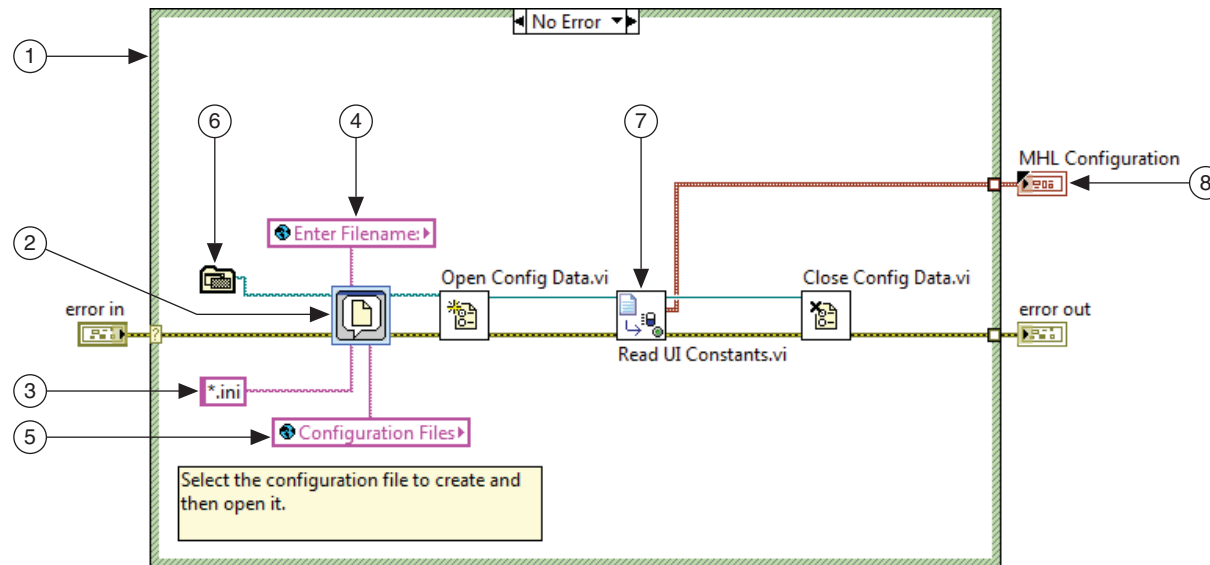
- Save the Read UI Constants VI.
- Add a new VI to the project under the **Support VIs»Configuration** virtual folder and save the VI as `Read Configuration Data.vi` in the `<Exercises>\LabVIEW Core 3\Course Project\support\Configuration` directory.
- Create the front panel and block diagram of the Read Configuration Data VI as shown in Figures 4-20 through 4-23. The Read Configuration Data VI reads the MHL configuration constants from the INI file and passes those values to the MHL.

Figure 4-20. Read Configuration Data VI Front Panel

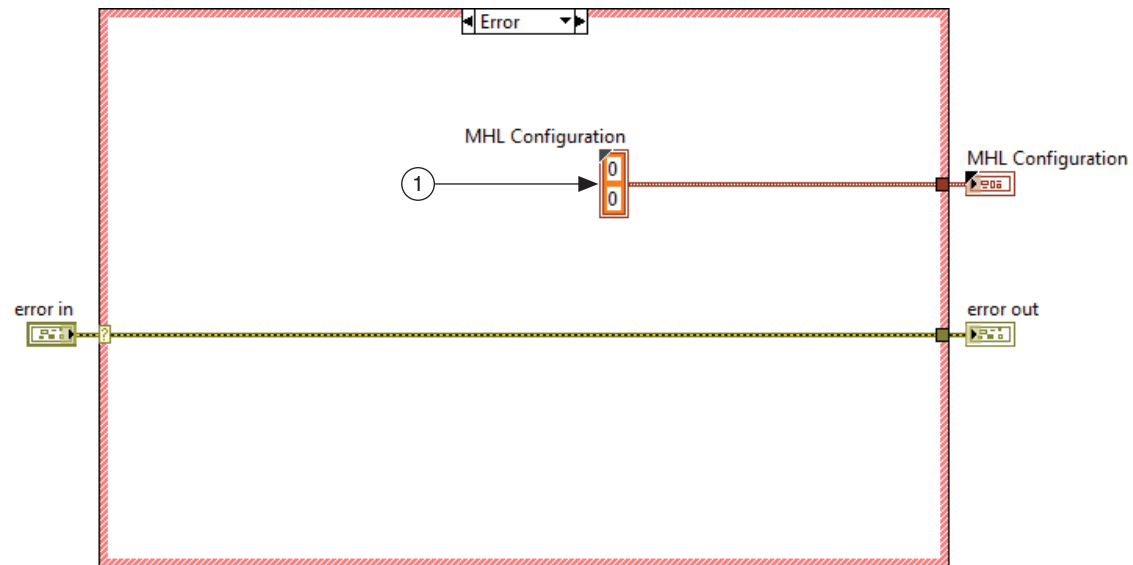


- 1 **Error In and Error Out**—Place an Error In and Error Out control and indicator on the front panel. After you complete the block diagram, the front panel will contain more items.

Figure 4-21. Read Configuration Data VI Block Diagram No Error Case

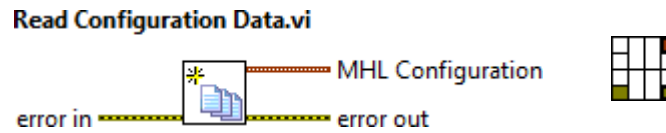


- 1 **Case Structure**—Place a Case structure on the diagram and wire the **error in** cluster to the case selector.
- 2 **File Dialog**—In the **Configure File Dialog** dialog box, place a checkmark in the **Limit selection to single item** checkbox and select the **File** and **New or existing** radio buttons. View the Express VI as an icon to save room on the block diagram.
- 3 String constant—Create a constant for the **pattern (all files)** input of the File Dialog Express VI and enter *.ini.
- 4 **Boiler System Globals variable**—Drag **Boiler System Globals.vi** from the **Project Explorer** window and select **Enter Filename** from the list. Right-click the global variable and select **Change To Read** from the shortcut menu. Wire this global variable to the **prompt** input of the File Dialog Express VI.
- 5 **Boiler System Globals variable**—Change the global variable to read and select **Configuration Files** from the list. Wire it to the **pattern label** input of the File Dialog Express VI.
- 6 **Application Directory**—Wire to the **start path** input of the File Dialog Express VI. Application Directory returns the path to the directory containing the application.
- 7 **Read UI Constants**—Drag this VI from the **Support VIs»Configuration** folder in **Project Explorer** window to the block diagram. You created this VI in step 3.
- 8 **MHL Configuration indicator**—Right-click the **MHL Configuration** output of the Read UI Constants VI and select **Create»Indicator**. Move the indicator outside of the Case structure and wire Read UI Constants VI to the indicator.

Figure 4-22. Create Read Configuration Data VI Block Diagram Error Case

1 **MHL Configuration constant**—Right-click the tunnel and select **Create»Constant** to create this constant.

Figure 4-23. Read Configuration Data VI Connector Pane

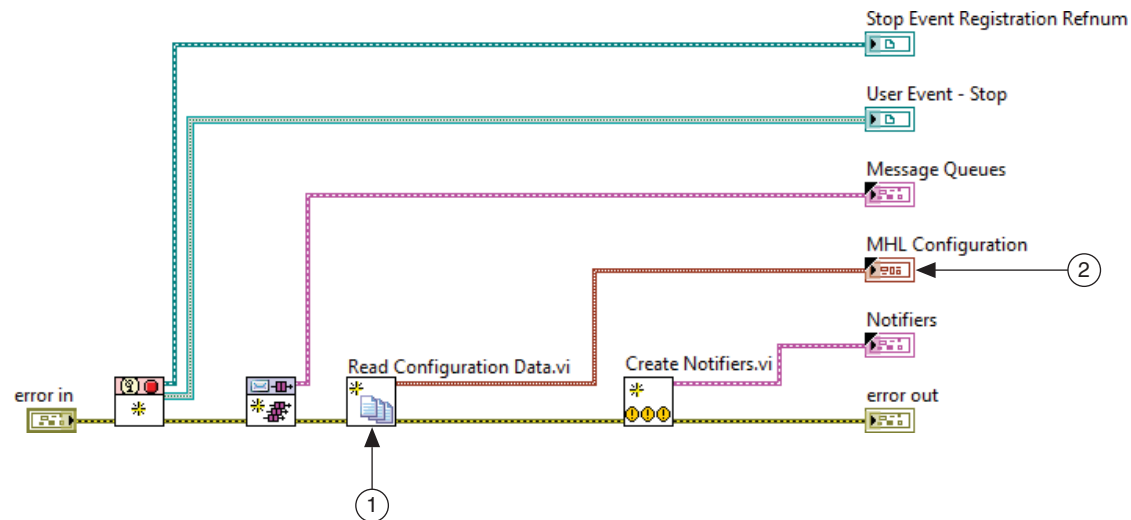


1 Filter glyphs by keywords, new, and file to find appropriate glyphs for the icon.

4. Update Boiler System Open.vi to include Read Configuration Data.vi.

- ☐ Open **Support VIs»Boiler System Open.vi** from the **Project Explorer** window.
- ☐ Update the VI as shown in Figure 4-24.

Figure 4-24. Boiler System Open VI including Read Configuration Data VI

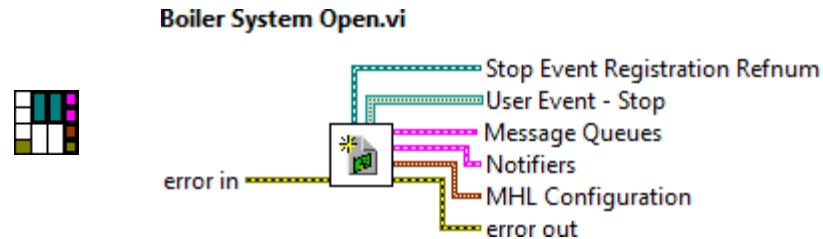


1 Read Configuration Data—Drag **Read Configuration Data.vi** from the **Project Explorer** window.

2 **MHL Configuration indicator**—Right-click the **MHL Configuration** output of the Read Configuration Data VI and select **Create»Indicator**.

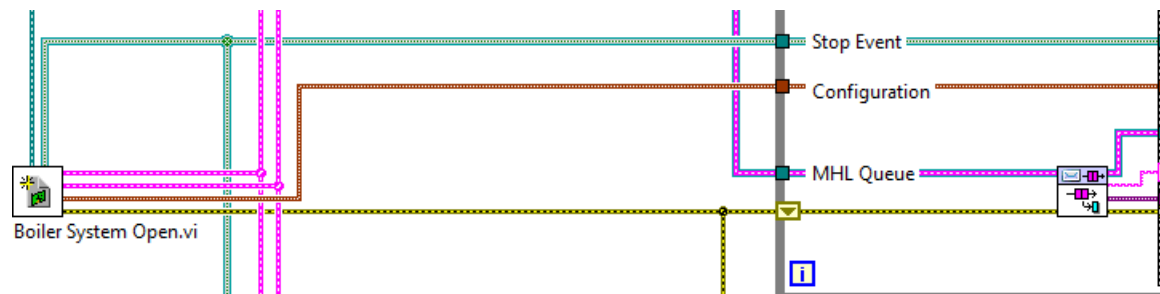
- Add MHL Configuration to the connector pane as shown in Figure 4-25.

Figure 4-25. Boiler System Open Icon and Connector Pane



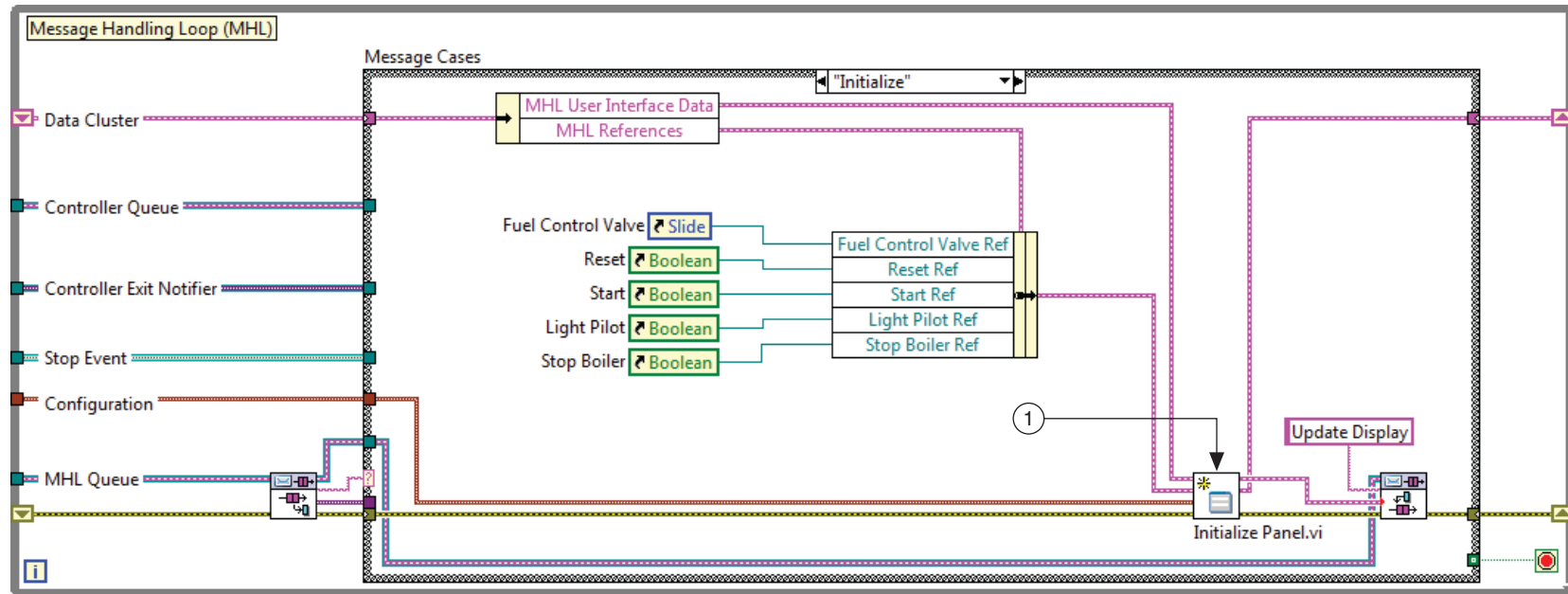
5. Add a VI to initialize the user interface, set the range for the Fuel Control Valve, and build the MHL Data cluster to the project.
 - Navigate to <Exercises>\LabVIEW Core 3\External\Support VIs and copy Initialize Panel.vi to <Exercises>\LabVIEW Core 3\Course Project\support.
 - Add Initialize Panel.vi to the **Support VIs** virtual folder in the Boiler Controller Project Explorer window.
6. Update the Main VI to read the configuration data and pass the data to the Initialize case of the MHL.
 - Wire the **MHL Configuration** output from the Boiler System Open VI to the MHL as shown in Figure 4-26.

Figure 4-26. Wiring the MHL Configuration Data to the MHL



- Update the Initialize case of the MHL as shown in Figure 4-27.

Figure 4-27. Updated Initialize Case of the Main MHL



1 **Initialize Panel**—Drag the Initialize panel VI from the **Project Explorer** window. Wire the Initialize Panel VI as shown.

- Save the Main VI.

Test the Application

Verify that previous functionality still works and that user interface events result in execution of the appropriate case in the boiler controller.

1. Test the VI.

- ☐ Run Main VI.
- ☐ Verify that a dialog prompts you to select an INI file.
- ☐ Select `Boiler Init.ini` in the `<Exercises>\LabVIEW Core 3\Course Project\` directory.
- ☐ Verify previous functionality still works, such as the display of one-button dialog boxes when you click the **Reset**, **Start**, or **Light Pilot** buttons.
- ☐ Verify the functionality of the **Fuel Control Valve** control.
 - The **Fuel Control Valve** control is initialized to 10.
 - The range of the **Fuel Control Valve** horizontal slider control is limited to values between 10 and 75.
 - Changing the value displays the one-button dialog box.

2. Update the build specification and test the new executable.

- ☐ Right-click the Main Application build specification and select **Properties** from the shortcut menu.
- ☐ Click **Source Files** in the **Category** list.
- ☐ Select **Boiler Init.ini** in the **Project Files** tree and move this file to the **Always Included** section.
- ☐ Click **OK** to save the build specification.
- ☐ Right-click the Main Application build specification and select **Build** from the shortcut menu.
- ☐ Click the **Explore** button when LabVIEW finishes the build.

☐ Double-click `Boiler Controller.exe` to run the application.

☐ Verify the behavior you tested in step 1.



Note The starting directory to browse to the INI file has changed. The build now creates the `Boiler Controller.ini`. This file is NOT the INI file that you created. `Boiler Init.ini` is inside the data folder.

☐ Click the **Emergency Stop** button and verify that all VIs stop and the application exits.

End of Exercise 4-3