

IVI™ VI Updater Utility User Guide

This document describes the purpose and use of the IVI VI Updater utility.

The Import CVI Instrument Driver Utility in LabVIEW converts a CVI version of an IVI driver to a LabVIEW version of an IVI driver. After conversion, you need to complete several steps to further update the driver functionality and create a more user-friendly driver interface.

Making these updates manually in each driver is time consuming and error prone. The IVI VI Updater utility performs many of the necessary tasks automatically, allowing you to quickly update and polish the converted LabVIEW VIs.



Note The IVI VI Updater utility has been created and optimized primarily for use with IVI drivers developed using the “Create IVI Instrument Driver...” wizard in LabWindows™/CVI™. In addition, the VIs updated using the IVI VI Updater utility require the presence of the IVI Engine 2.0 on the system. The IVI VI Updater utility has not been thoroughly tested with the *VXIplug&play* drivers to verify proper and expected behavior, function, or performance. When used with the *VXIplug&play* drivers, National Instruments (NI) does not warrant that the IVI VI Updater utility will meet your requirements or that the use or operation thereof will be uninterrupted or error-free.

Understanding the IVI VI Updater Utility

The utility applies different types of VI updates based on the type of driver VI being updated. The utility processes two types of driver VIs: class-defined VIs and instrument specific VIs.

Task Flowchart

The utility loads the driver VIs one by one. When one VI is loaded, the utility checks if the VI is a class driver VI, or a specific driver VI. Refer to Figure 1 for the complete task flow.

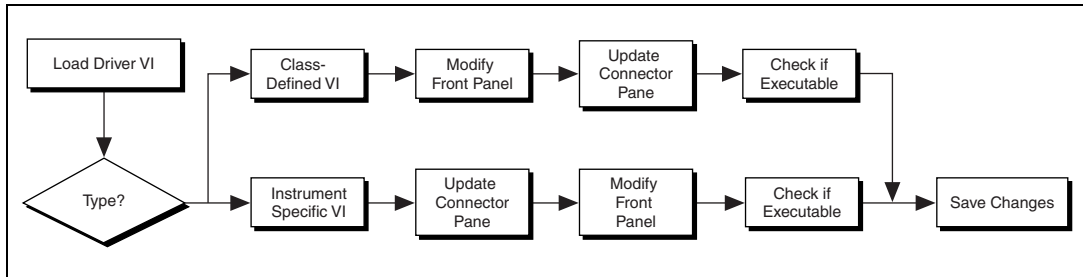


Figure 1. IVI VI Updater Task Flowchart

Class-Defined VIs

The utility uses class driver VIs as templates for updating class-defined VIs. The utility modifies the class-defined VIs as follows:

- Updates the label and caption
 - Changes the text to match class driver template VI
 - Uses the bold text style
 - Displays captions and hides labels

Each control has both a label and caption, and the caption should appear on the front panel rather than the label.

When matching a class-defined driver VI with a template VI, the utility changes the control label of the driver VI to match the control label used in the template VI. Differences between these names are reported in the Update Report Window, as shown in Figure 5.

- Appends the default value to the label and caption
 - Normal Controls—Searches for the default value in the control description
 - Enumerated—Gets the default value from item text
 - Boolean—Gets the default value from Boolean text

The utility modifies the text style and appends the control default value to the label and caption. If the control is an Enumerated or a Boolean, the utility directly gets the default value. For other types of controls, the utility looks for the default value from the control description. If the utility cannot find the default value, the utility displays a message in the Update Report Window, as shown in Figure 5.

- Applies engineering notation
The utility changes all the floating points controls and indicators to engineering notation with two digits of precision.
- Changes Boolean text to `True` or `False`
The utility changes the Boolean text to `True` or `False` for all Boolean controls and indicators.
- Changes the error code to use hex style
The utility makes both error in and error out clusters use hex display for the error code.
- Updates the front panel layout
 - Arranges control positions to match class driver template VIs
 - Arranges panel size to match class driver template VIs
 - Sets **Instrument Handle** terminal to `Required`
 The utility arranges the controls in the same position as the template, then modifies the panel size to match the template.
- Updates the connector pane
 - Sets the terminal pattern to match class driver template VIs
 - Sets the terminal links to match class driver template VIs
 - Sets **Instrument Handle** to `Required`
 The utility matches the connector pane to the template. The instrument handle connection is set to `Required`.
- Checks if the VI is executable
The utility checks if the specific VI can be executed. The utility displays a warning message in the Update Report Window, as shown in Figure 5, if the VI is not executable.

Instrument Specific Driver VIs

The utility modifies instrument specific VIs as follows:

- Updates the connector pane
 - Sets the terminal pattern to 4-2-2-4 or 5-2-2-2-5
 - Sets **Instrument Handle** to `Required`
 The controls are wired so that the terminal layout pattern matches the front panel layout pattern. The terminal pattern is set to either 4-2-2-4 or 5-2-2-2-5.
- Updates the label and caption
 - Uses the bold text style
 - Displays captions and hides labels

- Appends the default value to label and caption
 - Normal controls—Searches for the default value in the control description
 - Enumerated—Gets the default value from item text
 - Boolean—Gets the default value from Boolean text
- Applies engineering notation

The utility changes all the floating points controls and indicators to engineering notation with two digits of precision.
- Changes Boolean text to **True** or **False**

The utility changes the Boolean text to **True** or **False** for all the Boolean controls and indicators.
- Changes the error code to use hex style

The utility makes both error in and error out clusters use hex display for error code.
- Arranges the front panel layout—Arranges layout of controls and indicators on the front panel to match the order used in the terminal pattern

The front panel layout for instrument specific driver VIs is not based on a template. Instead, the utility lays out the front panel according to the following subtasks:

 - Places the instrument handle in the left top of the front panel; places the instrument handle indicator on the right top; places the error control on the left bottom; and places the error indicator on the right bottom
 - Sets a fixed space between each two rows or two columns
 - Locates controls to the left and locates the indicators to the right
 - Matches the front panel layout pattern of the controls to the terminal layout pattern
- Checks if the VI is executable

The utility checks if the specific VI can be executed. The utility displays a warning message in the Update Report Window, as shown in Figure 5, if the VI is not executable.



Note The Import CVI Instrument Driver utility in LabVIEW follows a hard-coded procedure for converting *VXIplug&play* function panels to LabVIEW VIs. The IVI functions such as `Initialize`, `Initialize With Options`, `Close`, `IVI Error Converter`, and `Utility` functions require further modifications. Use the IVI VI Updater utility to update these specific functions correctly.

Using the IVI VI Updater

To update a LabVIEW IVI driver, complete the following steps:

1. Launch the utility from **Start»Programs»National Instruments»IVI Tools**.

The utility has the following command buttons at the bottom of the front panel:

Help

Configure

< Back

Next >

Cancel

- **Help**—Click this button to access this help document for the utility
- **Configure**—Opens the configuration window
- **Back**—Click to go back to the previous step
- **Next**—Click to go to the next step
- **Cancel**—Click to exit the utility

2. Click **Configure** to launch the configuration window. The configuration window has five tabs: General, IVI Driver, VXIp&p Driver, Front Panel, and Other, as shown in Figure 2.

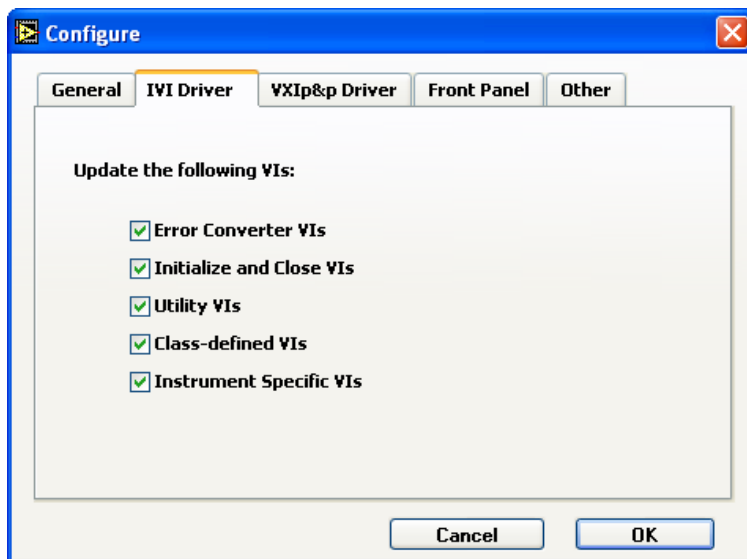


Figure 2. Configuration Window: IVI Driver Tab

3. On the **General** tab, select the version of LabVIEW you want to use to save the updated VIs.

4. Select which VIs you want to update on the **IVI Driver** tab, as illustrated in Figure 2. For more information about how the utility updates VIs, refer to the *Class-Defined VIs* section and the *Instrument Specific Driver VIs* section. If you are updating a *VXIplug&play* driver, click the **VXIp&p Driver** tab.
5. Click the **Front Panel** tab. Select which updates you want to make on the front panel controls and indicators. The following items are selected on the Front Panel tab.
 - a. Modify layout
 - b. Modify controls and indicators. Choose from the following list of controls and indicators:
 - Label and caption
 - Floating point control
 - Ring control
 - String control
 - Boolean control
 - “instrument handle” control
 - “Error code” control
 - Other controls
6. Click the **Other** tab, and select the following options:
 - a. Update the connector pane
 - b. Check if the VI is executable
 - c. Change absolute DLL path to DLL name
7. Click **OK**.
8. Click **Next**.

- Use the IVI Class drop down menu to select the IVI class of your driver. In Figure 3, IVI Function Generator is selected.

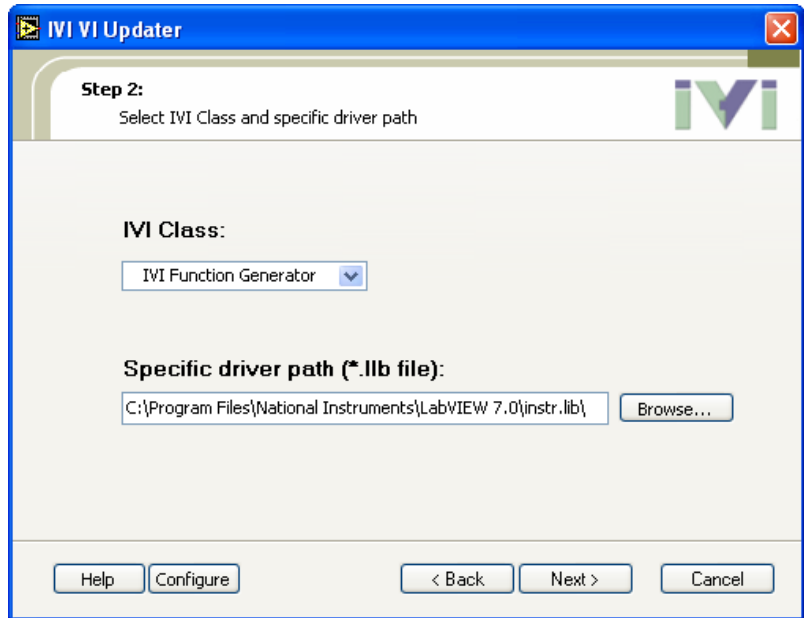


Figure 3. Select IVI Class and Specific Driver Path Window

The utility references a template and matches it to your driver. Eight classes are included in the utility: DC Power Supply, Digital Multimeter (DMM), Function Generator, Oscilloscope, Power Meter, RF Signal Generator, Spectrum Analyzer, and Switch.



Note If your driver is *not* class compliant, select **IVI Custom**.

- Input the full path of your driver LLB file in **Specific driver path**, or browse to the location of the driver and click **Open**, as shown in Figure 4.
- Click **Next**.
- Review the specific driver prefix and DLL name before continuing. If the wrong information is displayed, enter the correct information.
- Click **Next**.

14. Use the provided Select, Deselect, Select All, and Deselect All buttons to select which VIs you wish to update, as shown in Figure 4.

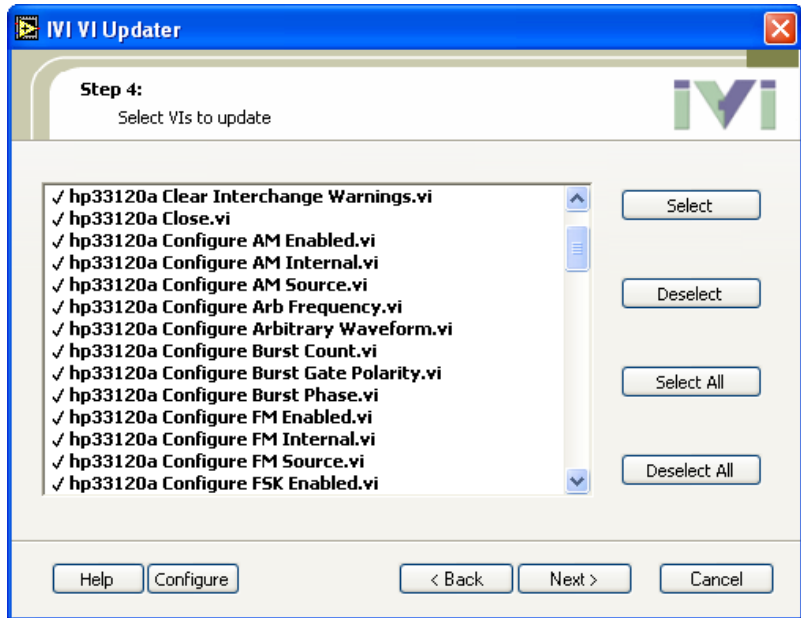


Figure 4. Select VIs to update Window

15. Click **Next**.

The progress bar shows the estimated time to completion and indicates which VI the utility is currently updating.

When the utility finishes, it displays an update report, as shown in Figure 5.

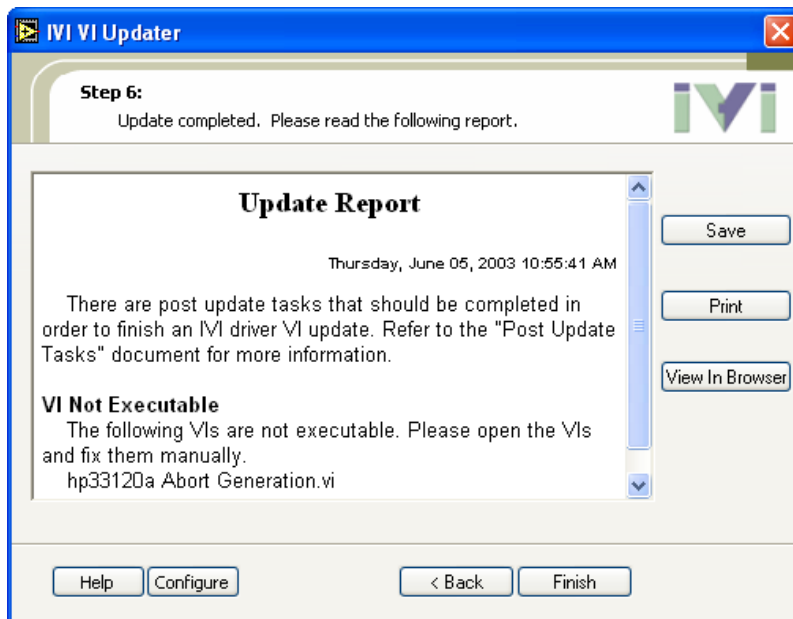


Figure 5. Update Report Window

Completing Post Update Tasks

The IVI VI Updater does not perform several necessary post-conversion update tasks. Therefore, even though the utility eliminates many manual steps, you still need to complete a few manual tasks. After you run the IVI VI Updater utility, complete the following tasks:

1. Read the messages in the update report window. A warning message includes the VI name and the message content. You may see any of the following warnings:
 - **Too many controls**—The utility uses either 4-2-2-4 or 5-2-2-2-5 patterns for the connector pane. If there are too many controls on a VI front panel, these two patterns are no longer suitable. You need to open the VI and modify the front panel and the connector pane manually.
 - **VI not executable**—If the driver VI cannot execute, the utility displays the message **VI not executable**. To solve this problem, open and fix the VI. Most likely, the utility could not find a subVI or DLL file.

- **Cannot find default value**—If the utility cannot find the default value for a specific control, it displays **Cannot find default value** in the update report window. Find the control and manually find and set the default value.
- **Control Name Change**—For class-defined VIs, the control name of the driver VI may be different than the template. In this case, the utility changes the control name; however, a warning message notifies you which control was changed. Verify that the new name is appropriate for your driver.
- **“Size of...” Control**—Each driver VI should be checked for both numeric and string array size controls. If a VI has array size controls, remove the Array Size parameters by completing the following tasks:
 - a. Check the block diagram for arrays passing into and out of the Call Library node. If there is an array, delete the size of the array control that is passed into the Initialize Array function and replace with the array size control that is passed directly into the Call Library node. Refer to Figures 6 and 7 for before and after examples.

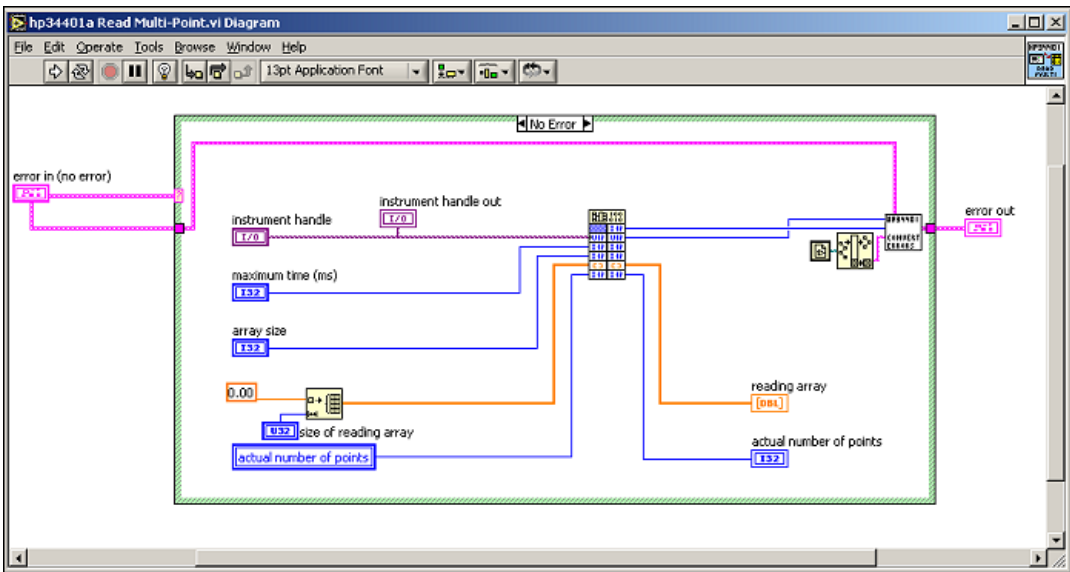


Figure 6. Array Size Manual Update—Before

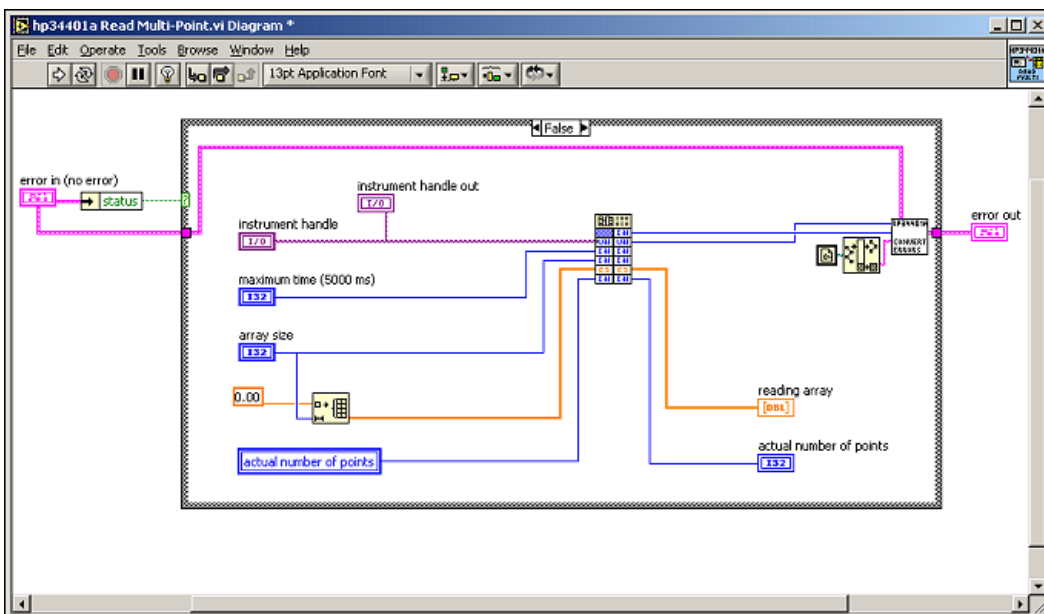


Figure 7. Array Size Manual Update—After

- b. For VIs with string indicators, replace the string size control with a block diagram constant. Select **256; 512; 1,024; 2,048;** and so on, as appropriate, for the string size. If the VI is an IVI-defined VI, you can look at the constant value used in the class driver block diagram for the appropriate value to use.
 2. For instrument specific driver VIs, make sure the VI has an appropriate icon. If the icon is incorrect, complete the following steps:
 - a. For instrument specific VIs, create or copy an icon.
 - b. Make the background light blue by copying the color from other VIs.
You can find icon examples in the Labview\Examples\Instr directory.
 - c. Make the top 1/3 of the icon the prefix name.
 3. For application example VIs and getting started VIs, complete the following steps:
 - a. Create application example VIs and getting started VIs, as needed, to help users with high-level operations, creating at least one application example VI and one getting started example VI. For class-complaint IVI drivers, look at existing IVI drivers for examples of these high-level VIs.
 - b. Add descriptions to the application example VIs and getting started example VIs.

- c. Edit the menu palette by adding the application example VIs and getting started VIs to the Application Functions subpalette.



Note Some users might choose to view the menu palettes in icon mode. Therefore, National Instruments recommends that the subpalettes also have useful icons. You can copy these subpalette icons from the icons used for the class drivers.

4. For all VIs, review each driver VI for other additional changes you might find necessary.

Giving Feedback

If you have comments or questions about this utility, contact National Instruments at `instrument.driver@ni.com`.

