WHEN YOU NEED TO BE SURE

CyFlex® LabView Interface Setup and Reference

Version 1

October 11, 2022

Developed by SGS North America, Inc.

Version History

Version	Date	Revision Description
1	10/11/2022	Initial version

Document Conventions

This document uses the following typographic and syntax conventions.

- Commands, command options, file names or any user-entered input appear in Courier type. Variables appear in Courier italic type.

  Example: Select the `cmdapp-`*`relVersion-buildVersion`*`.zip` file….

- User interface elements, such as field names, button names, menus, menu commands, and items in clickable dropdown lists, appear in Arial bold type.

  Example: **Type**: Click **Select Type** to display drop-down menu options.

- Cross-references are designated in Arial italics.

  Example: Refer to *Figure 1*…

- Click intra-document cross-references and page references to display the stated destination.

  Example: Refer to *Section 1 Introduction on page 1.*

  The clickable cross-references in the preceding example are *1*, *Introduction*, and on page 1.

Related Documents

CyFlex manuals are available at https://cyflex.com/. View **Help & Docs** topics or use the **Search** facility to find topics of interest.

Table of Contents

List of Figures

1 Introduction

This document describes setup of an interface between CyFlex® and LabView that enables LabView to access CyFlex workspace and the resulting capabilities of this interface.

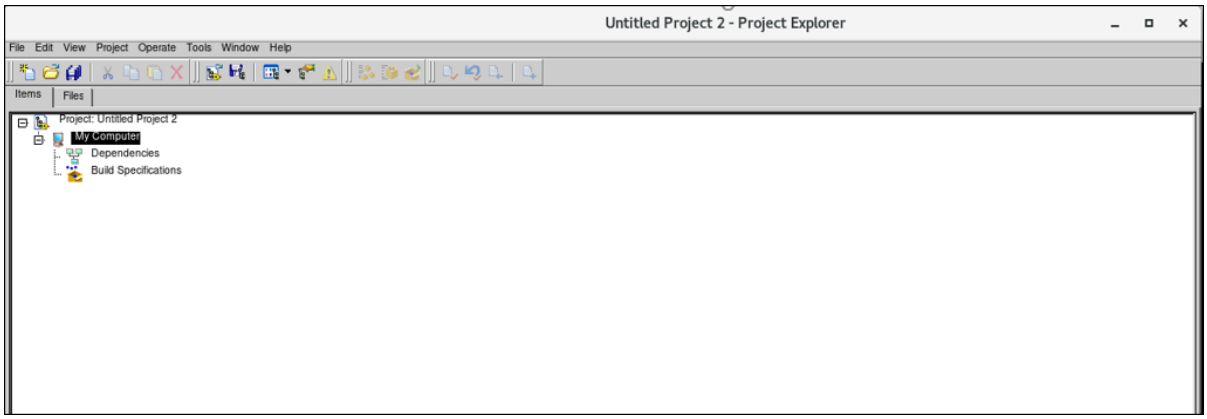
Included are details about the library and function calls available to LabView, as well as a practical example for each call.

2 Setting Up a LabView Project

Execute the following steps in LabView:

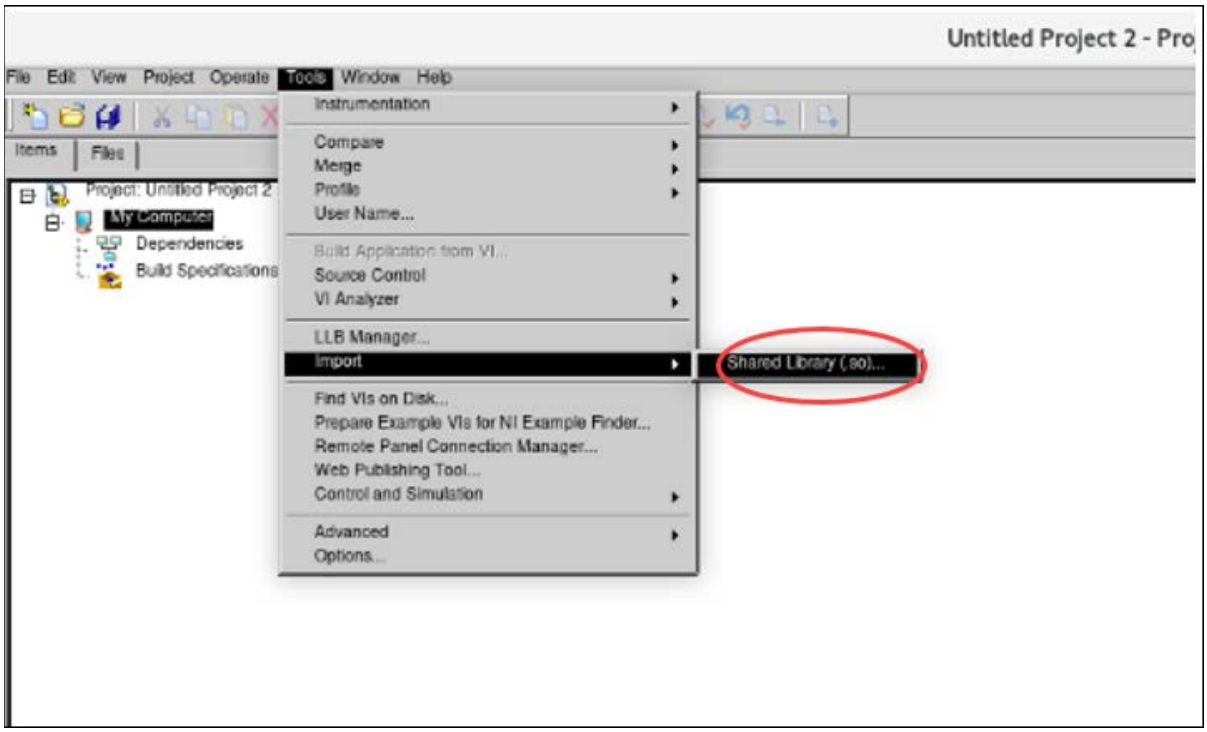
1. Open an existing project file, file type `.lvproj`, or create a new project. *Figure 1* shows an example of creating a new, untitled project.

Figure 1: New Untitled Project



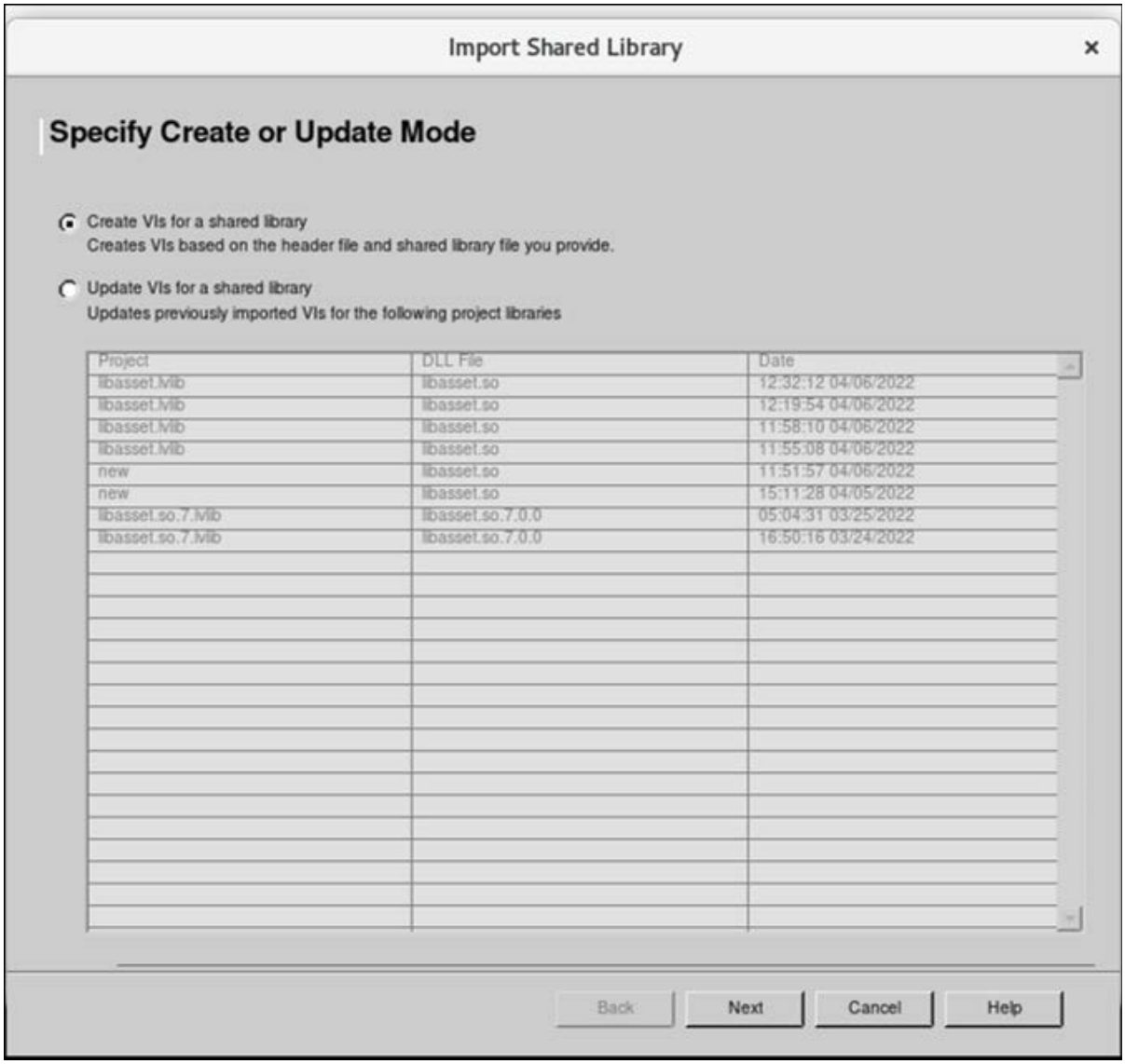
2. Click **Tools** on the Menu Bar, hover the mouse over the sub-menu **Import** entry, and then select **Shared Library (.so)** as indicated in *Figure 2*.

Figure 2: Select Shared Library (.so)



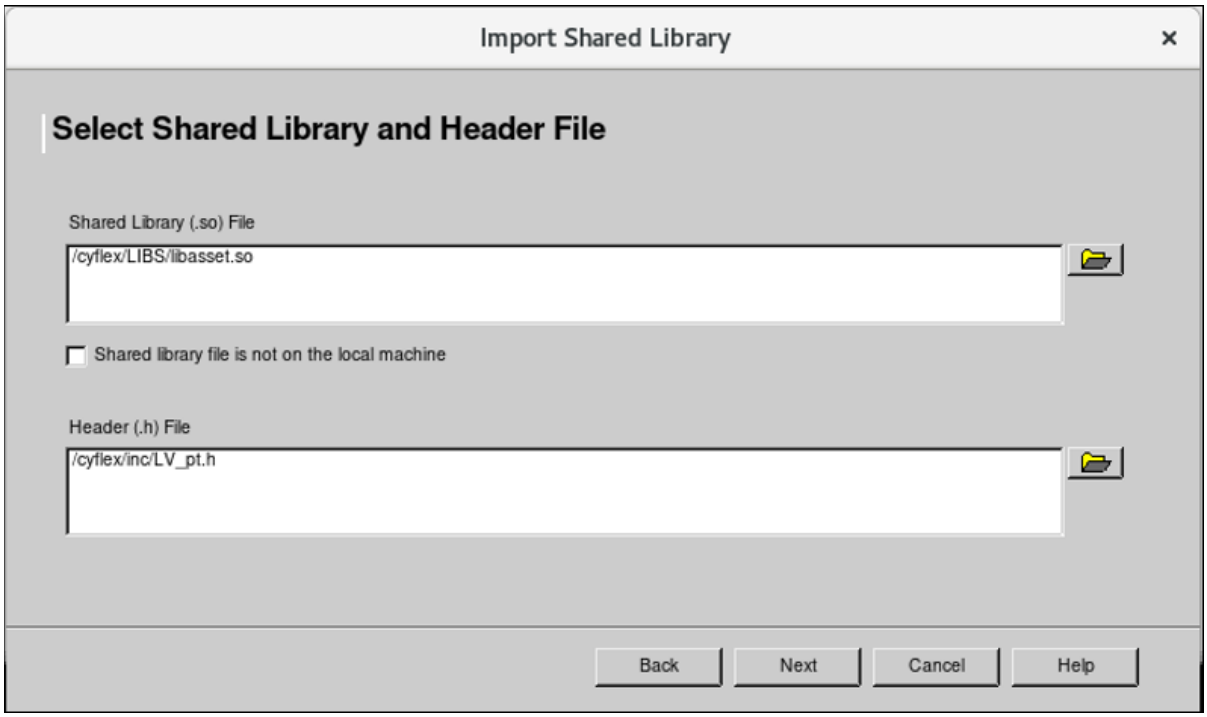
3. On the resulting screen, select **Create VI for a shared library** to **Specify Create or Update Mode** and then select **Next** as in *Figure 3.*

Figure 3: Import Shared Library Selections



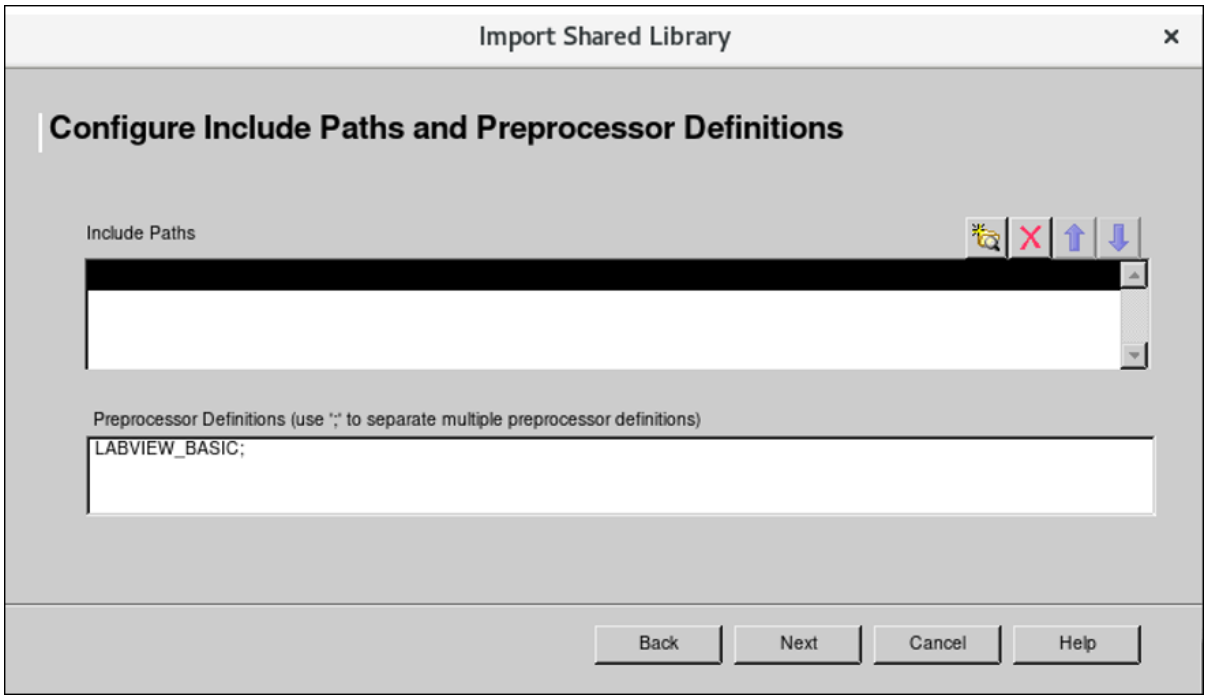
4. **Select Shared Library and Header File** for the Imported Shared Library. Enter `/cyflex/LIBS/libasset.so` library and `/cyflex/inc/LV_pt.h` header file as in *Figure 4* and then select **Next**.

Figure 4: Shared Library and Header File Entries



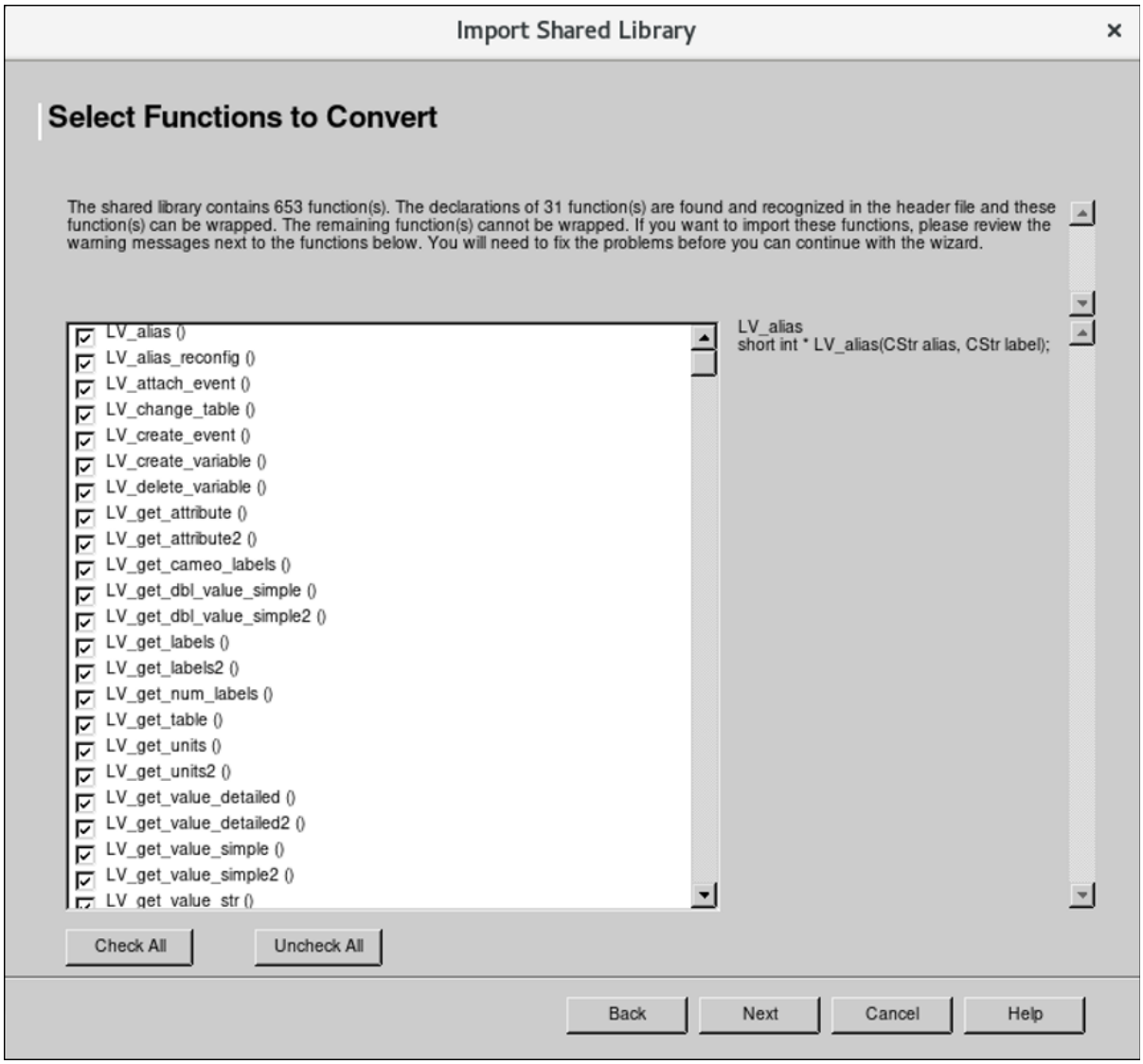
5. To **Configure Include Paths and Preprocessor Definitions**, enter `LABVIEW_BASIC:` in the **Preprocessor Definitions** box and then select **Next** as in *Figure 5*.

Figure 5: Configure Include Path and Preprocessor Definition



6. The resulting window as in *Figure 6* lists the available functions. The LV_ appended functions are automatically selected. These functions will be utilized to access the CyFlex workspace variables. No action is required on this screen other than to click **Next**.

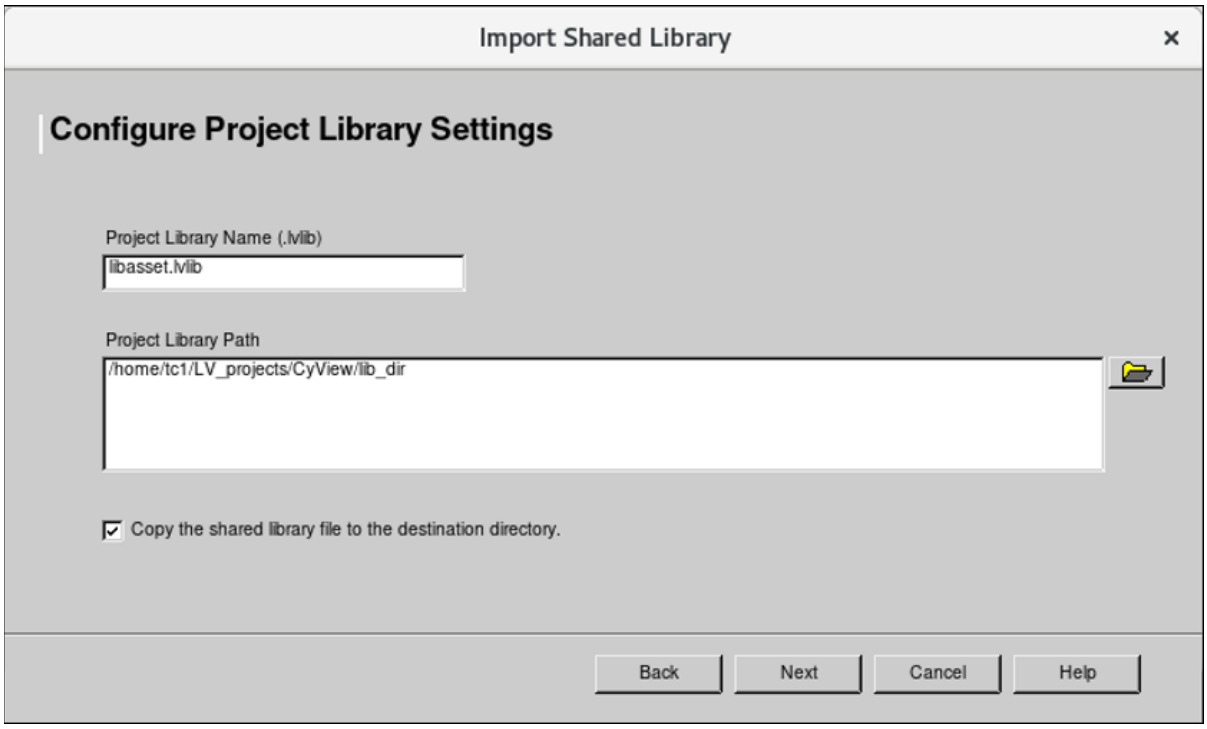
Figure 6: Select Functions to Convert



7. **Configure Project Library Settings**. on the next screen. Enter a **Project Library Name** and **Project Library Path** as in the example in *Figure 7*. The path must be located in an empty user-created folder that does not require higher privileges. A directory under the home directory fulfills this requirement.

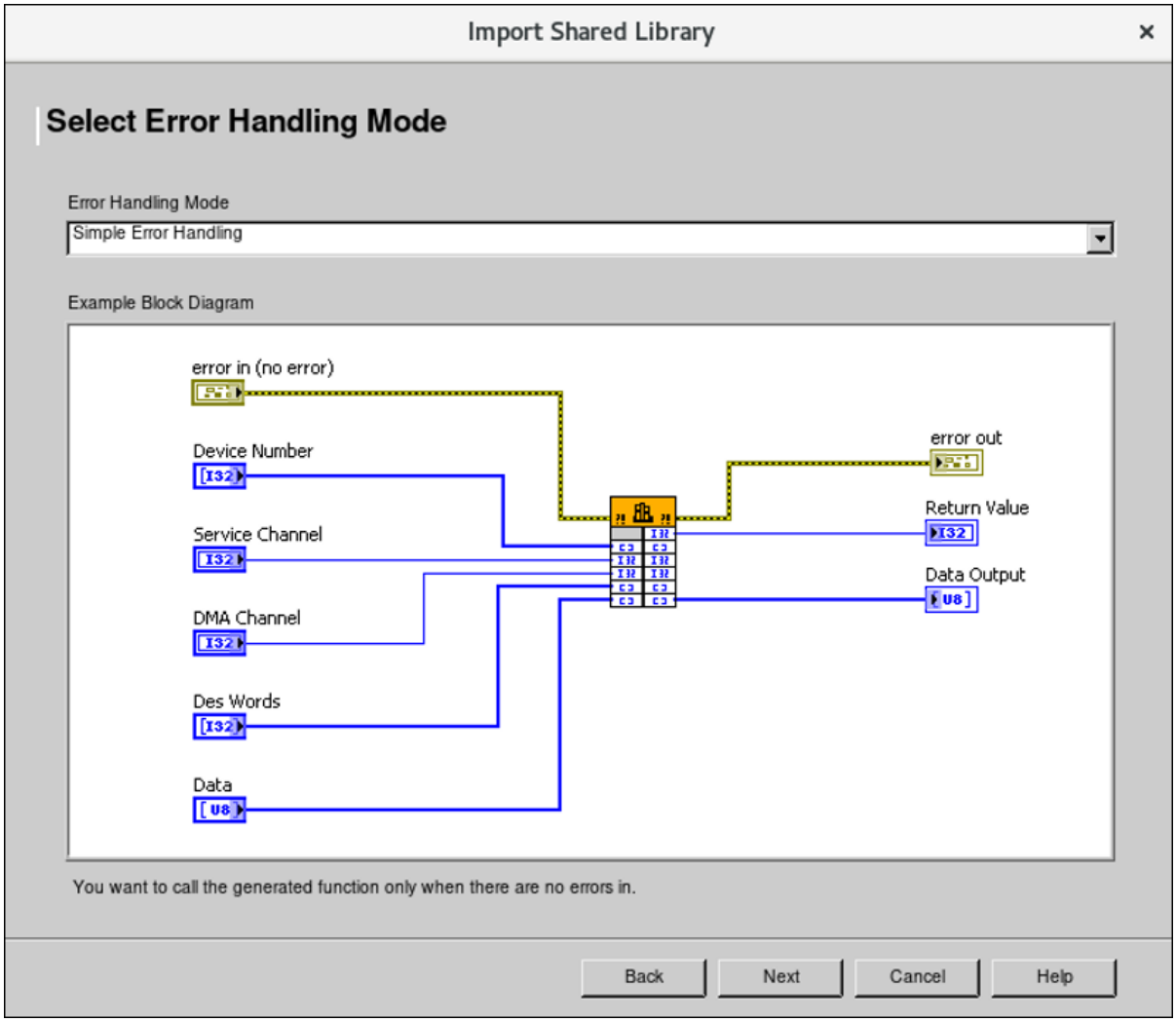
   Accept the default selection to **Copy the shared library file to the destination directory** and then select **Next**.

Figure 7: Configure Project Library Settings



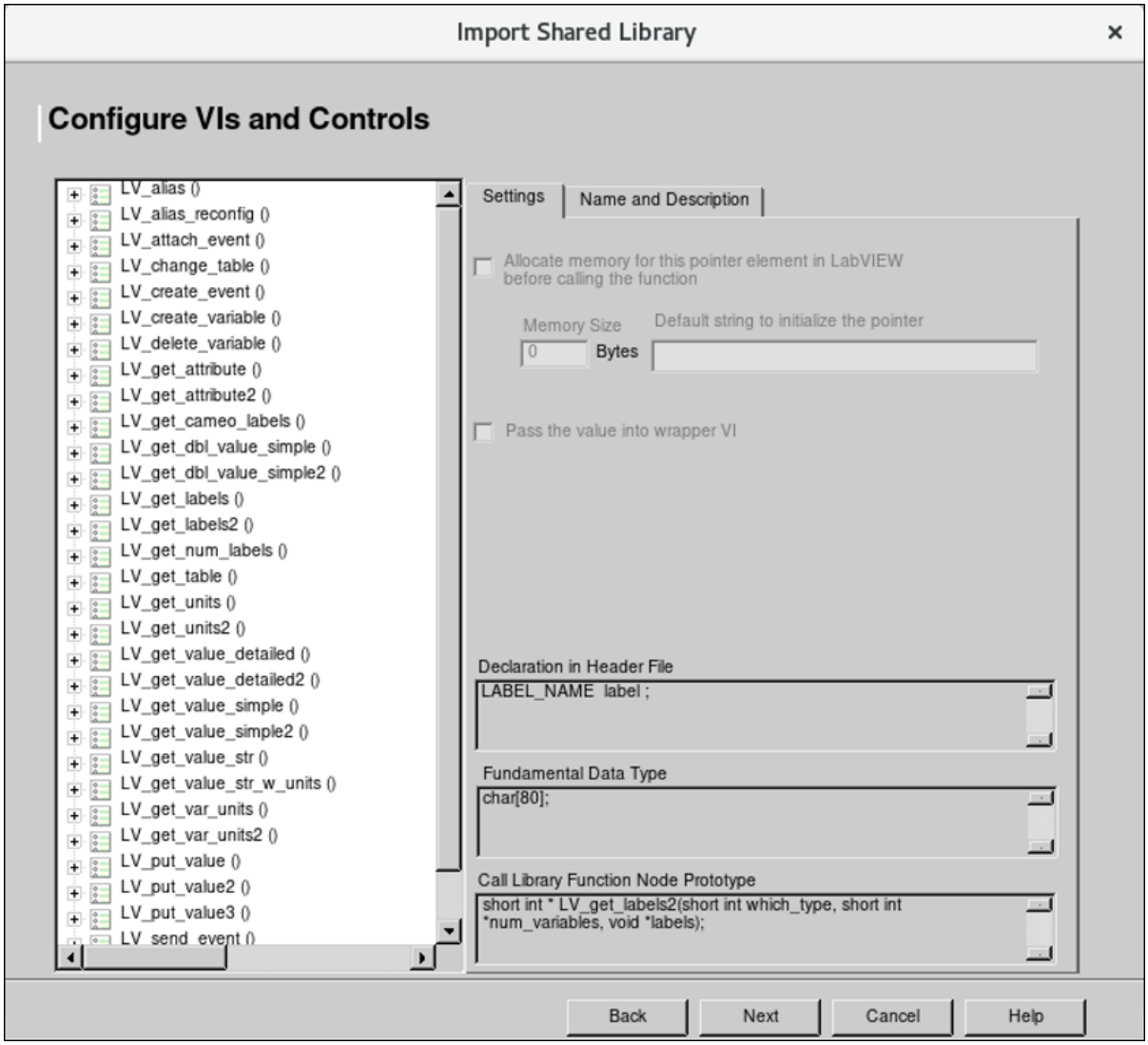
8. **Select Error Handling Mode**. Click the pull-down and select **Simple Error Handling** as in *Figure 8* and then select **Next**.

Figure 8: Select Error Handling Mode



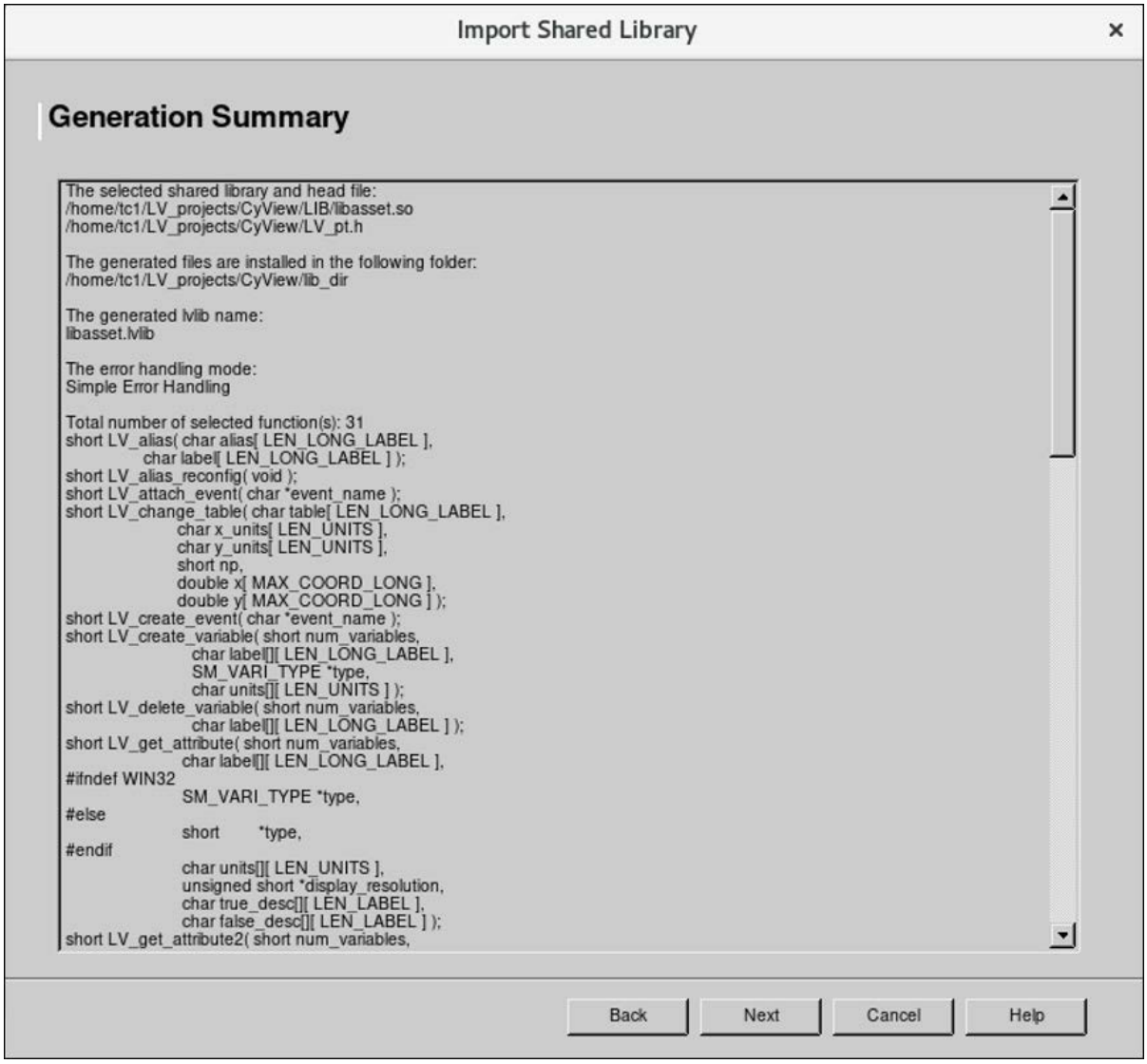
9. The **Configure VIs and Controls** screen as in *Figure 9* displays additional configuration options. No action is required other than to select **Next**.

Figure 9: Configure VIs and Controls



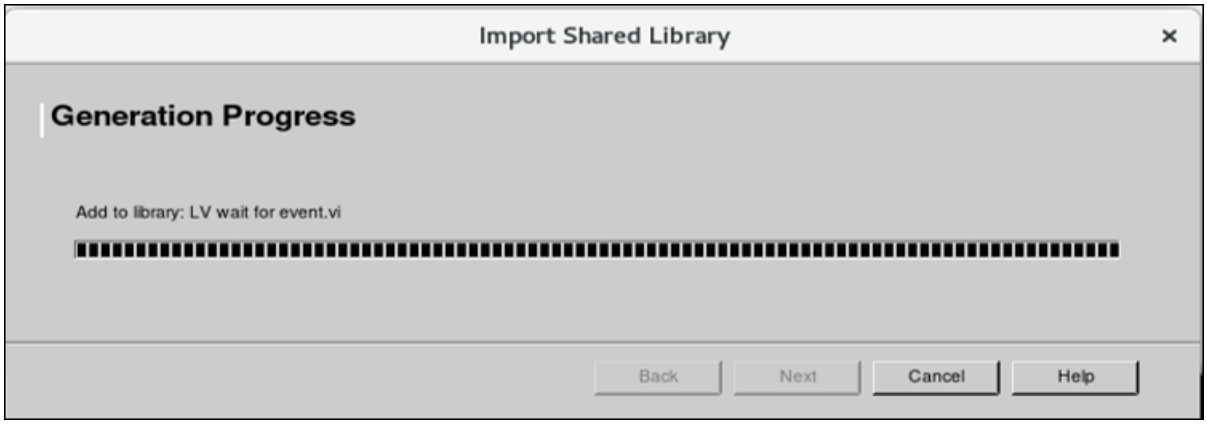
10. A summary is generated and displayed as in *Figure 10*. Review as needed and then click **Next**.

Figure 10: Generation Summary



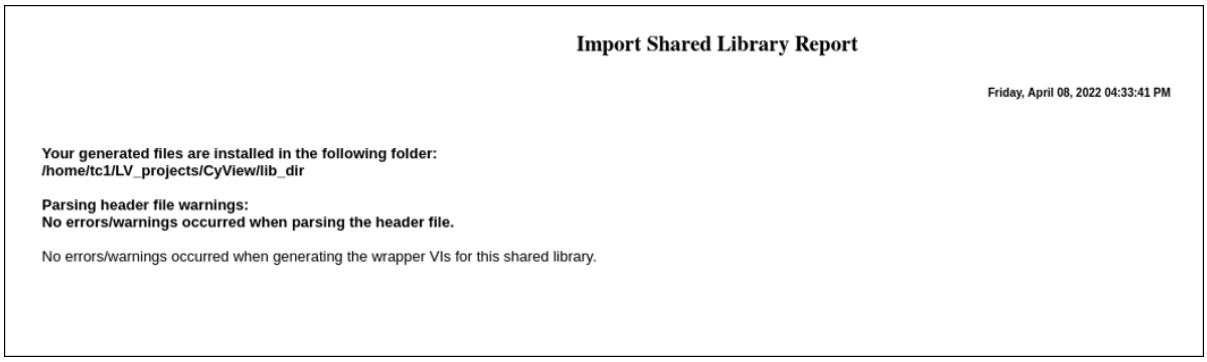
11. VI Generation commences. **Generation Progress** is displayed as in *Figure 11*. Click **Next** when generation completes.

Figure 11: Generation Progress



12. A summarized **Import Shared Library Report** is displayed as in *Figure 12*. Review the files in the mentioned folder as needed.

Figure 12: Import Shared Library Report



3 Library Functions and Capabilities

The library provides the capability to create, query, set, and respond to CyFlex events and numerous types of CyFlex variables.

This section lists available functions.

3.1 Event Functions

3.1.1 LV_send_event

Functional Description:	Enables LabView to set events in the CyFlex environment
Input	Event_name [STRING]
Outputs	• Message {STRING] • Event Name [STRING] • Function Return [DBL]
Notes	The event must exist before calling this function.

3.1.2 LV_create_event

Functional Description:	Creates an event in the CyFlex Environment
Input	Event_name [STRING]
Outputs	• Event Name [STRING] • Function Return [DBL]
Notes	N/A

3.2 Variable Functions – Retrieve Type

3.2.1 LV_get_attribute

Functional Description:	Enables LabView to obtain detailed information about a CyFlex variable
Inputs	• Number Of Variables [DBL] • Label [STRING]
Outputs	• Function Return [DBL] • Label [STRING] • Type Code [DBL] • Units [STRING] • Display Resolution [DBL] • True Description [STRING] • False Description [STRING]
Notes	N/A

3.2.2 LV_get_value_simple

Functional Description:	Enables LabView to obtain only the value of a variable
Inputs	• Number Of Variables [DBL] • Label [STRING]
Outputs	• Function Return [DBL] • Value [DBL]
Notes	N/A

3.2.3 LV_get_value_str

Functional Description:	Enables LabView to retrieve a CyFlex string variable's value
Inputs	• Number Of Variables [DBL] • Label [STRING]
Outputs	• Function Return [DBL] • Label [STRING] • String Value [STRING]
Notes	N/A

3.2.4 LV_get_value_detailed

Functional Description:	Enables LabView to obtain detailed information about a CyFlex variable including the value
Inputs	• Number Of Variables [DBL] • Label [STRING] • Units [STRING]
Outputs	• Function Return [DBL] • Label [STRING] • Value [DBL] • Type Code [DBL] • Units [STRING] • Last Update [DBL]
Notes	N/A

3.2.5 LV_get_units

Functional Description:	Enables LabView to obtain information about CyFlex units
Input	• Unit [STRING]
Outputs	• Function Return [DBL] • Input Unit [STRING] • Num of Unit Codes in that Unit Group [DBL] • SI Unit [STRING]
Notes	N/A

3.2.6 LV_get_num_labels

Functional Description:	Enables LabView to obtain information for a specific type code
Input	• Type Code [DBL]
Outputs	• Function Return [DBL] • Number of Labels [DBL]
Notes	This function will return the number of variables that are of the specified type code.

3.2.7 LV_get_var_units

Functional Description:	Enables LabView to obtain the units of a specified variable
Inputs	• Number of Variables [DBL] • Label [STRING]
Outputs	• Function Return [DBL] • Label [STRING] • Units [String]
Notes	N/A

3.2.8 LV_get_dbl_value_simple

Functional Description:	Enables LabView to return the value of CyFlex variable without units
Inputs	• Number of Variables [DBL] • Label [STRING]
Outputs	• Function Return [DBL] • Label [STRING] • Value [DBL]
Notes	N/A

3.3 Variable Functions – Setting Type

3.3.1 LV_put_value

Functional Description:	Enables LabView to set the value of a CyFlex variable
Inputs	• Number of Variables [DBL] • Type Code [DBL] • Value [DBL] • Label [STRING] • Units [STRING]
Outputs	• Function Return [DBL] • Label [STRING] • Units [STRING] • Type Code [DBL] • Value [DBL]
Notes	N/A

3.3.2 LV_delete_variable

Functional Description:	Deletes a CyFlex variable from the CyFlex workspace
Inputs	• Number of Variables [DBL] • Label [STRING]
Outputs	• Function Return [DBL] • Label [STRING]
Notes	N/A