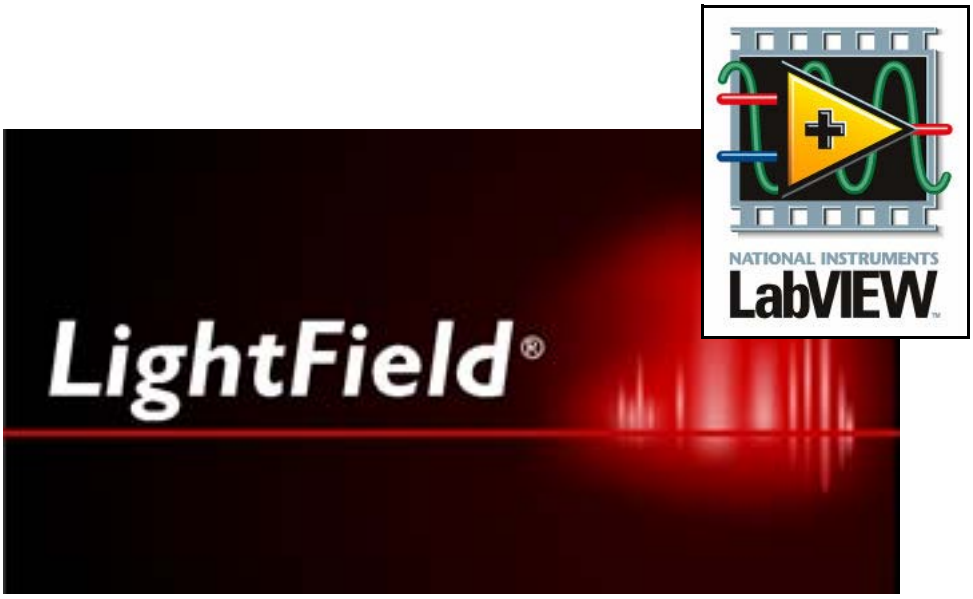
Introduction to Automating LightField 5 with LabVIEW®



Princeton Instruments

4411-0150
Issue 3
August 29, 2018

Revision History

Issue	Date	List of Changes
Issue 3	August 29, 2018	Issue 3 of this document incorporates the following changes: • Corrected typos.
Issue 2	January 2, 2018	Issue 2 of this document incorporates the following changes: • Updated the copyright year.
Issue 1	June 9, 2015	This is the initial release of this document.

Princeton Instruments, a division of Roper Scientific, Inc.
3660 Quakerbridge Rd
Trenton, NJ 08619
TEL: 800-874-9789 / 609-587-9797
FAX: 609-587-1970

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction to Automation

LightField is able to interface with several third-party applications in order to automate the configuration and performance of experiments within LightField.

Automation is the ability to use a programming environment (e.g., LabVIEW®,) to control LightField externally and performing such tasks as:

- Modifying experiment configuration parameters during such as:
  - Exposure Time
  - Frames to Save
- Connecting/disconnecting hardware
- Displaying live data as it is being acquired
- Importing previously acquired data that have been saved

Automation differs from LightField Add-in in that LightField is actually being controlled by an external program as opposed to running what is, effectively, a LightField subroutine.

This document provides an introduction to developing automation routines using National Instruments' LabVIEW® programming environment.

NOTE:

This document is not intended to be a tutorial about using LabVIEW®. It is written with the assumption that the reader possesses a basic knowledge of the LabVIEW® environment.

This document provides information about the following:

1.1 Prerequisites

In order to automate LightField using LabVIEW®, the following requirements must be satisfied:

- LightField 5 must be installed with the Add-Ins and Automation SDK option included.

NOTE:

For complete information, refer to the installation instructions included with LightField.

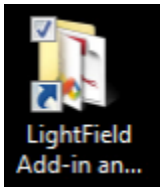
- LabVIEW® 2013 (or later) must be installed.

This page is intentionally blank.

Chapter 2: Prepare LightField

This chapter describes the steps necessary to prepare LightField for automation using LabVIEW®.

2.1 Verify SDK Installation



Before developing any automation routines, verify the **Add-Ins and Automation SDK** has been included as part of the LightField installation.

When the SDK has been included, LightField places a shortcut on the host computer desktop. This shortcut points to a directory in which information required by the automation application developer is stored. This information includes:

- LightField Add-ins and Automation Programming Manual.pdf

  Provides detailed programming information about the API and dotNET function required to develop an automation application.
- Experiment XML Specification.pdf

  This is the specification that defines how a LightField experiment file is structured.
- SPE 3.0 File Format Specification.pdf

  This is the specification that defines how LightField saves acquired data.

  In order to access data after it has been saved, how it is stored/save must first be understood which is detailed in this document.
- LightField Experiment Settings.chm

  This Windows Help File provides information about all settings used when interacting with LightField. The information included in this help file is required when developing automation applications.

If the shortcut is not located on the host computer's desktop, it is a good indication that the SDK has not been included in the current LightField installation. Therefore, uninstall LightField and reinstall it with the custom option to add the **LightField Add-in and Automation SDK** selected.

2.2 Create Default Experiment

It is recommended that a baseline, default LightField experiment be created and saved prior to beginning any automation application development. Doing so will reduce the number of preliminary and initial configuration steps required by the automation application.

Perform the following procedure to create a default experiment that will serve as the basic experiment for automation routines:

1. Launch LightField.
2. Once LightField has finished initializing, verify that all devices are properly installed and have been placed on the Device Grid within the Experiment Workspace.
3. Configure a baseline set of experiment parameters.

   These should be appropriate for the specific experiment and application for which an automation application is to be developed.
4. Once the baseline configuration is set, save the experiment with a descriptive name that will be easily recognizable for future use. The name of this file will be required as an input when developing the LabVIEW® automation application.

Chapter 3: Build a Simple Automation Application

Once a preliminary LightField project template has been created and saved as described in Chapter 2, Prepare LightField, on page 7, development of an automation application can begin.

NOTE:

This document is not intended to be a tutorial about using LabVIEW®. It is written with the assumption that the reader possesses a basic knowledge of the LabVIEW® environment.

Perform the following procedure to develop a simple automation application that will:

- Launch LightField
- Modify the experiment exposure time
- Close LightField.

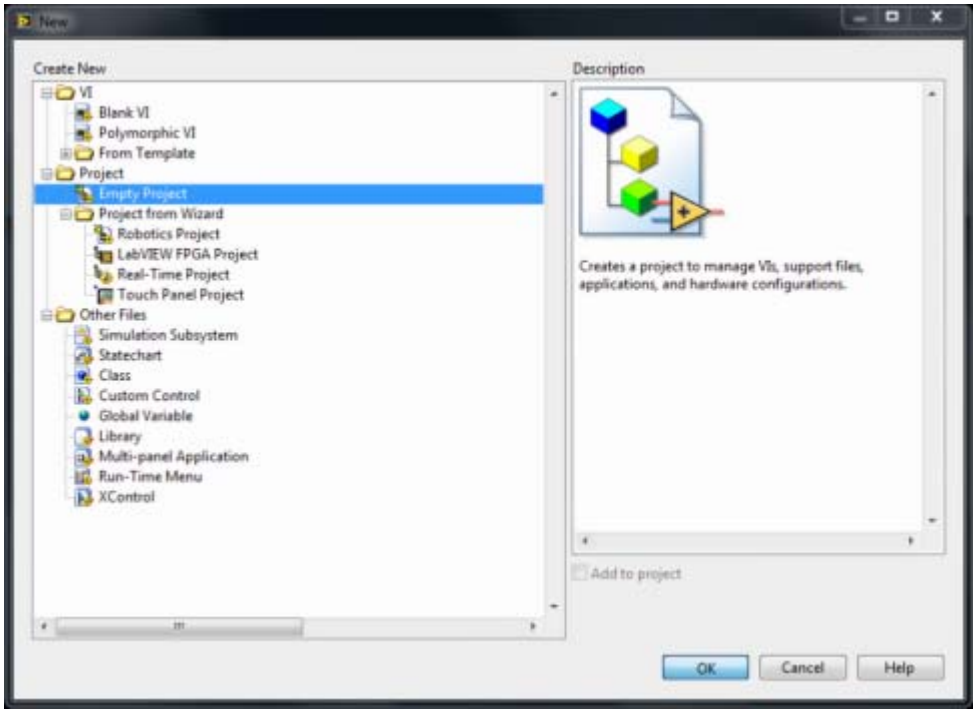
1. Launch LabVIEW 2013.
2. From the primary LabVIEW window, select **File —> New…** to create a new LabVIEW project. See Figure 3-1.

Figure 3-1: Primary LabVIEW Window

4411-0150_001

3. Within the New Project window, expand the **Project** folder (if necessary,) highlight **Empty Project**. See Figure 3-2.

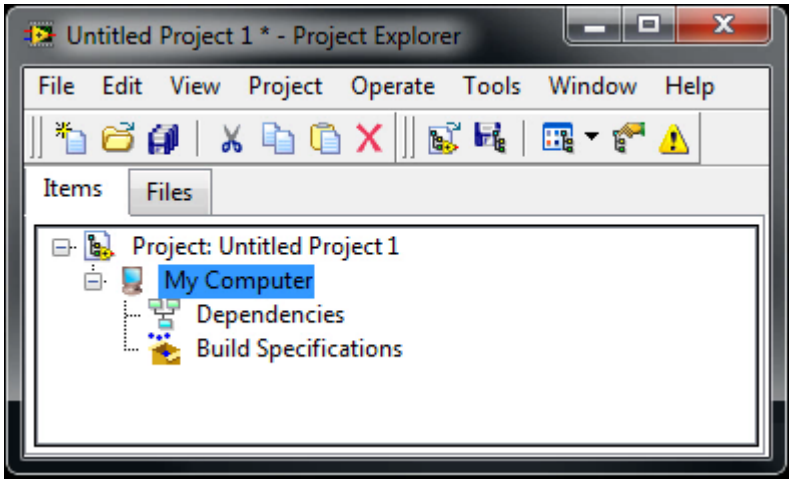
Figure 3-2: Typical LabVIEW New Project Window



4411-0150_002

Click **OK** to create a new, empty project. See Figure 3-3.

Figure 3-3: Typical Empty LabVIEW Project



4411-0150_003

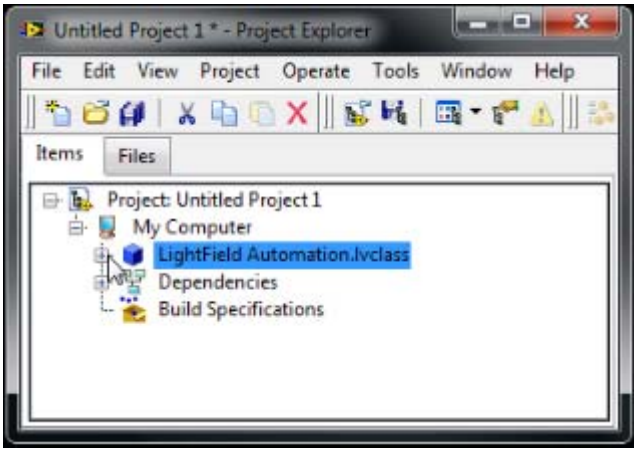
4. On the host computer's desktop, open the **Add-Ins and Automation SDK** folder by clicking on the shortcut that LightField placed there.
5. Open the **Samples** folder.
6. Open the **LabVIEW Automation** folder.

7. Open the **lightfield-automation-labview** folder.

   Within this folder is a file named `LightField Automation.lvclass`. This is a LabVIEW class file that contains all methods required for automating LightField.

8. Click on `LightField Automation.lvclass` and drag it into the Project Explorer for the newly created empty LabVIEW project. See Figure 3-4.

Figure 3-4: Adding `LightField Automation.lvclass` to a New Project Explorer

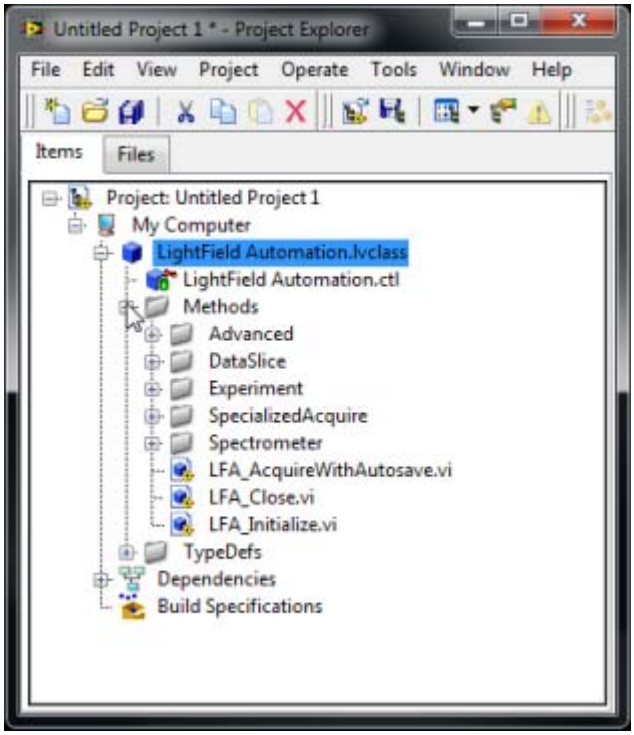


4411-0150_004

Once the `LightField Automation.lvclass` is in place, the **Add-Ins and Automation SDK** folder can be closed/dismissed.

9. Next, click on the **+** next to the `LightField Automation.lvclass` to expand it. Once expanded, open the **Methods** folder to view each of the items available within it.

Figure 3-5: Expanded `LightField Automation.lvclass` Object



4411-0150_005

10. Within the **Project Explorer**, select **File —> New VI** to begin creating the new automation application.

    Two workspaces are opened:

    - The Front Panel

      This serves as the final automation application's Graphical User Interface. All necessary controls are found here once the application has been developed, acquired data are displayed here, and additional GUI styling can be provided if desired.

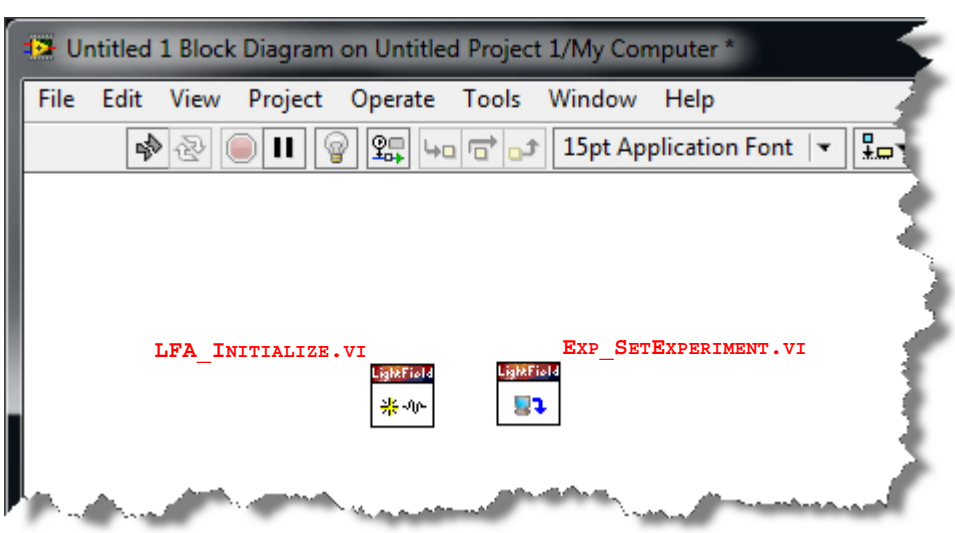
    - The Block Diagram

      This is where the automation application is developed and built using standard "building blocks" which are created and logically "connected."

11. If necessary, click in/on the **Block Diagram** panel to bring it into focus.
12. Within the **Methods** folder on the **Project Explorer** panel:
    a. Locate the `LFA_Initialize.vi` method, and drag it into the **Block Diagram**.
    b. Open the **Experiment** folder, and drag the `Exp_SetExperiment.vi` method into the **Block Diagram**.

    See Figure 3-6.

Figure 3-6: Beginning to Build an Automation Application

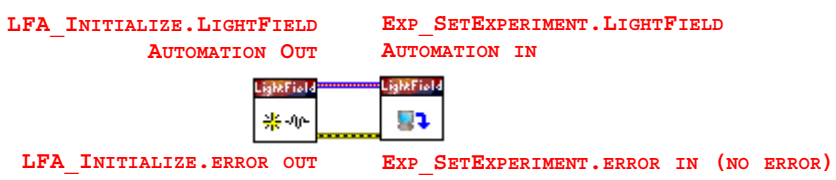


13. On the **Block Diagram**, connect `LFA_Initialize.error out` to `Exp_SetExperiment.error in (no error)`. See Figure 3-7.

Figure 3-7: Error Control Connections

14. On the **Block Diagram**, connect `Exp_SetExperiment.LightFieldAutomation in` to `LFA_Initialize.LightField Automation out`. See Figure 3-8.

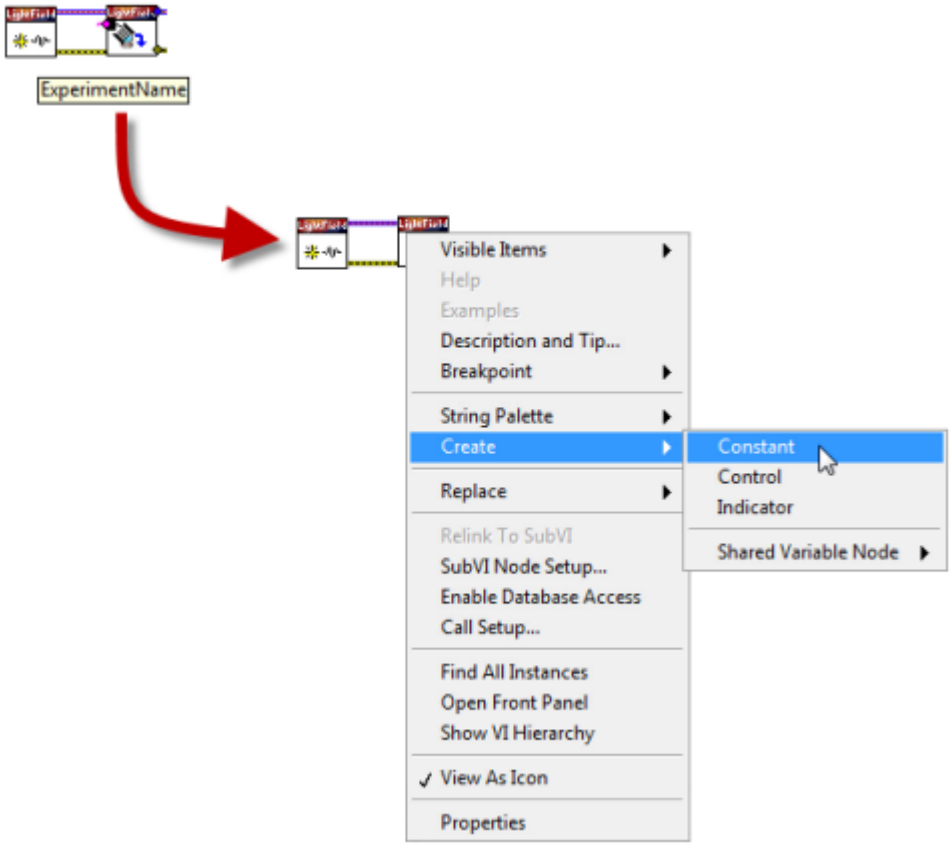
Figure 3-8: Initialize Automation Connections



4411-0150_008

15. Click on the `SetExperiment.ExperimentName` input control, and select C**reate —> Constant** from the pop-up menu displayed.

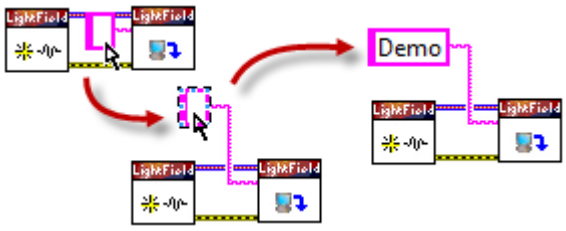
Figure 3-9: Define Experiment Input Control



4411-0150_009

16. Double click on the newly created `Experiment Name` constant field and enter the name of the experiment that is to be loaded by the Automation application (e.g., **Demo**.) See Figure 3-10.

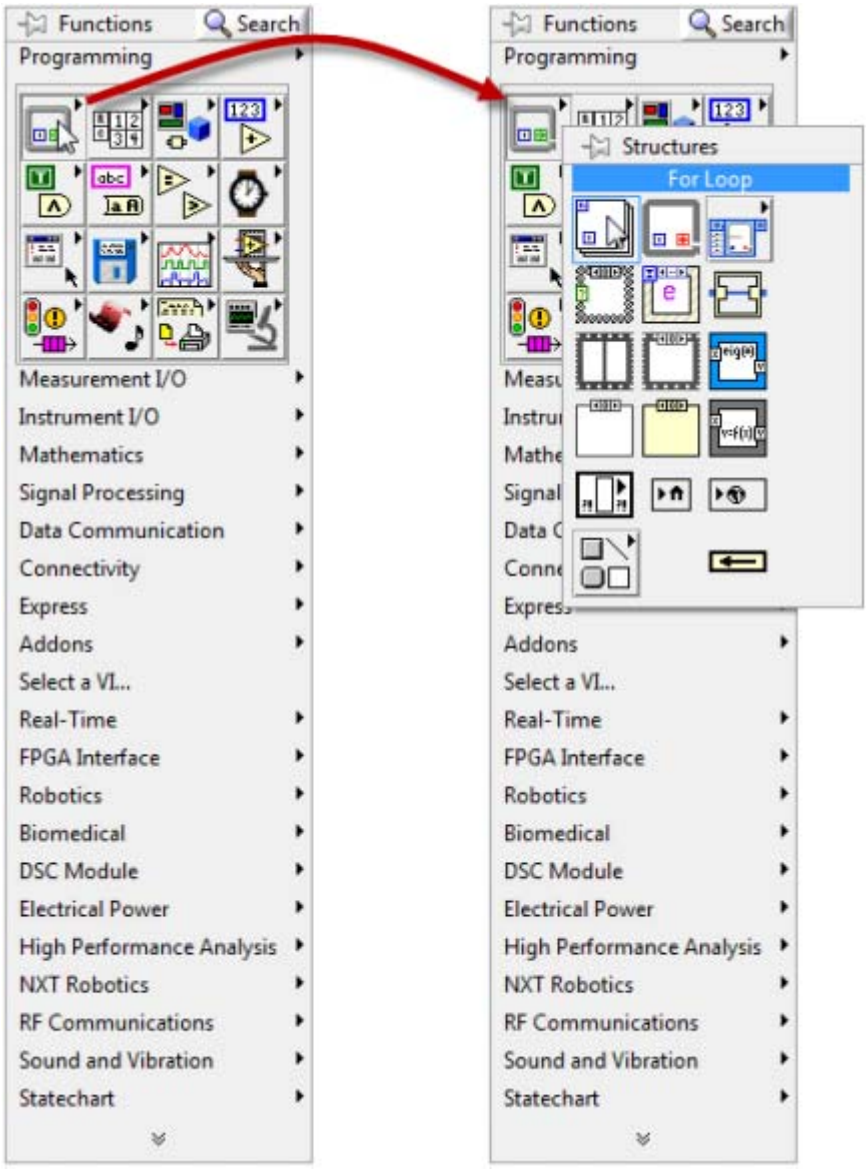
Figure 3-10: Specifying the Input File to be Loaded



4411-0150_010

17. Click within the **Block Diagram** to view the Functions menu panel similar to that shown in Figure 3-11.

Figure 3-11: Functions Menu Panel



18. Click on the Structures button  in the upper left-hand corner of the menu panel to view the Structures menu.

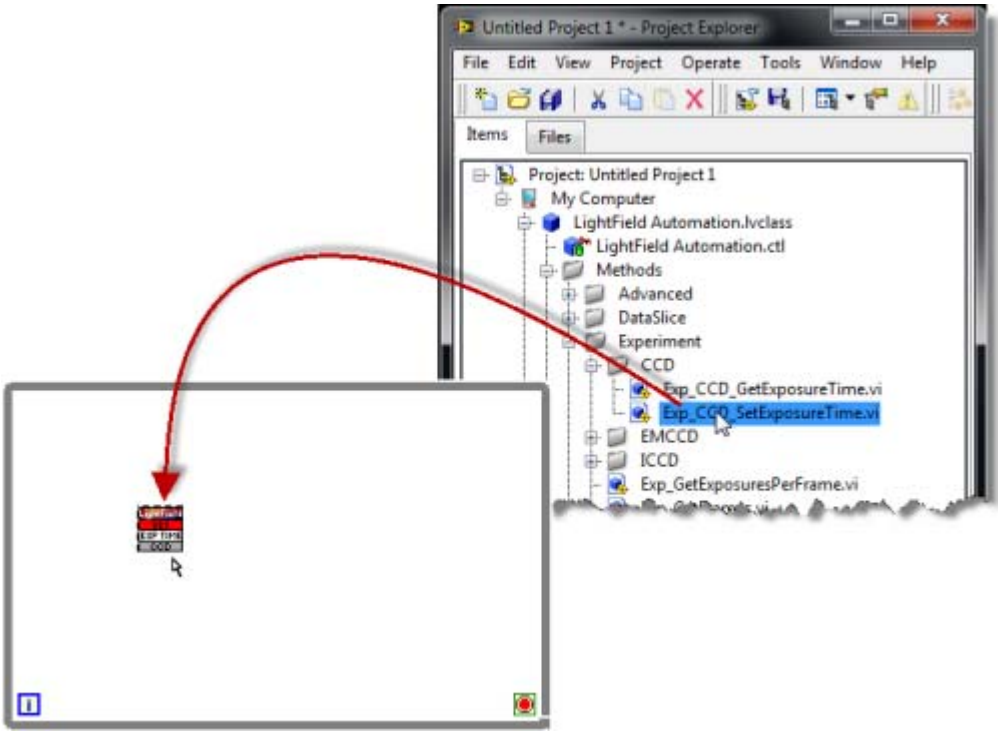
19. Click on the `While Loop` structure  and within the **Block Diagram** drag a rectangle to create a `While Loop` control box. See Figure 3-12.

Figure 3-12: While Loop Control Box

4411-0150_012

20. Within the **Project Explorer**, expand the **Experiment —> CCD** folder (if necessary.) Locate the `Exp_CCD_SetExposureTime.vi` method and click/drag it into the `While Loop` control object.

Figure 3-13: Adding the SetExposureTime.vi Object

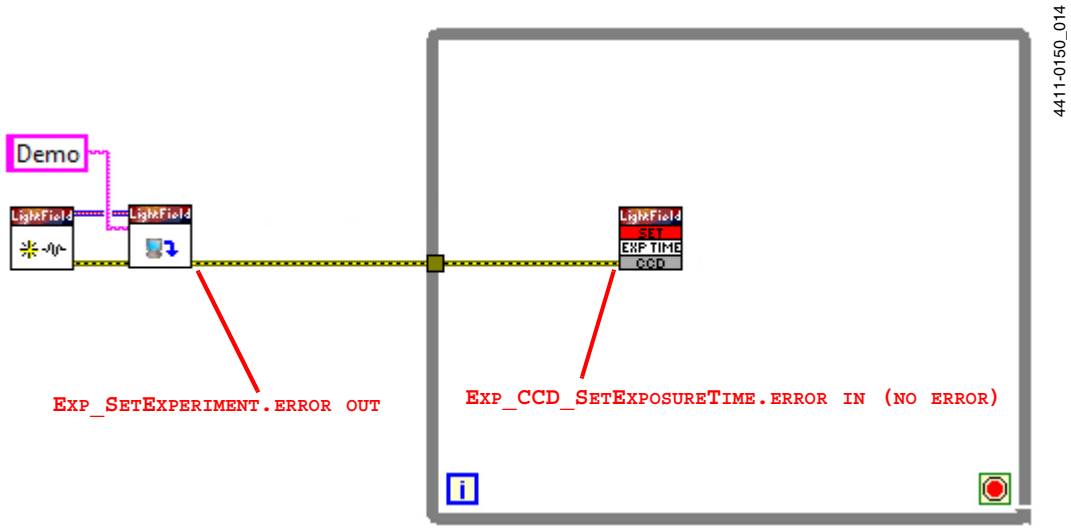


4411-0150_013

WHILE LOOP
CONTROL OBJECT

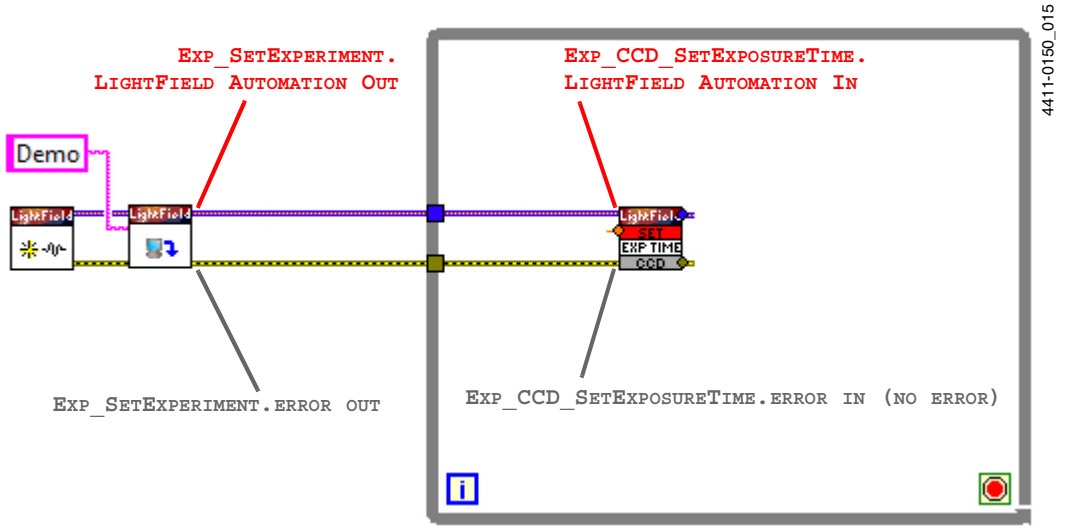
21. On the **Block Diagram**, connect `Exp_SetExperiment.error out` to `Exp_CCD_SetExposureTime.error in (no error)`. See Figure 3-14.

Figure 3-14: Set Exposure Time Error Control Connections



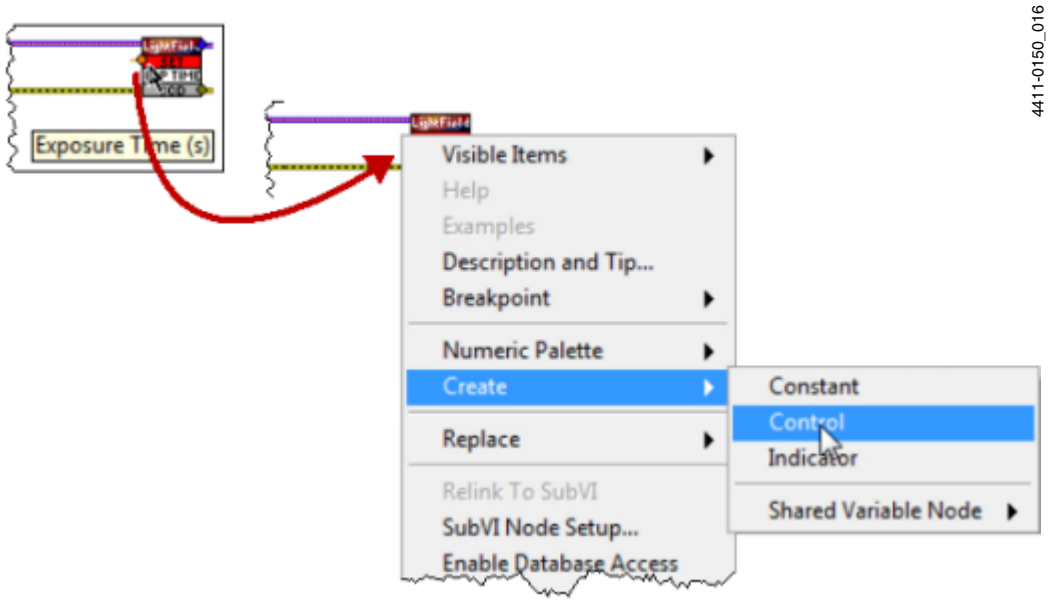
22. On the **Block Diagram**, connect `Exp_SetExperiment.LightField Automation Out` to `Exp_CCD_SetExposureTime.LightField Automation In`. See Figure 3-15.

Figure 3-15: Set Exposure Time Automation Control Connections



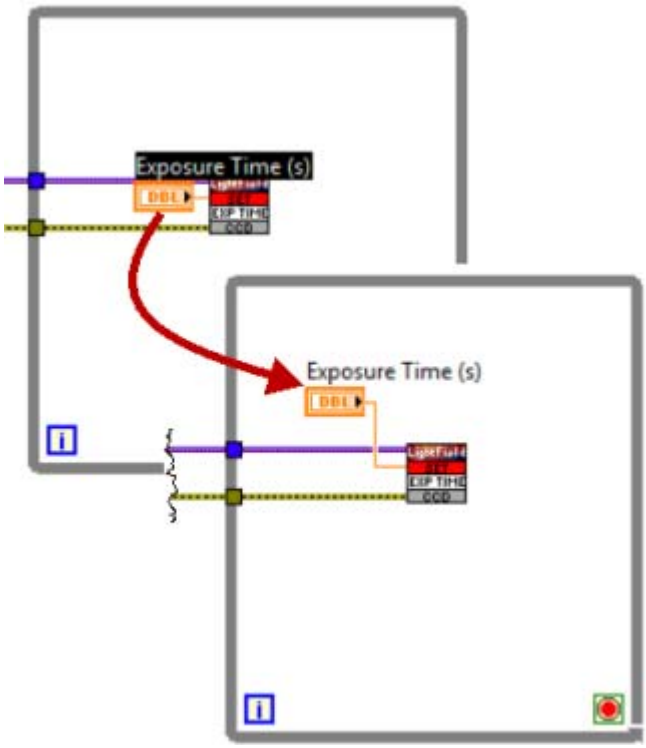
23. On the **Block Diagram**, click on the `Exposure Time (s)` control and select **Create —> Control** from the pop-up menu displayed to create a new control element. See Figure 3-16.

Figure 3-16: Create Control



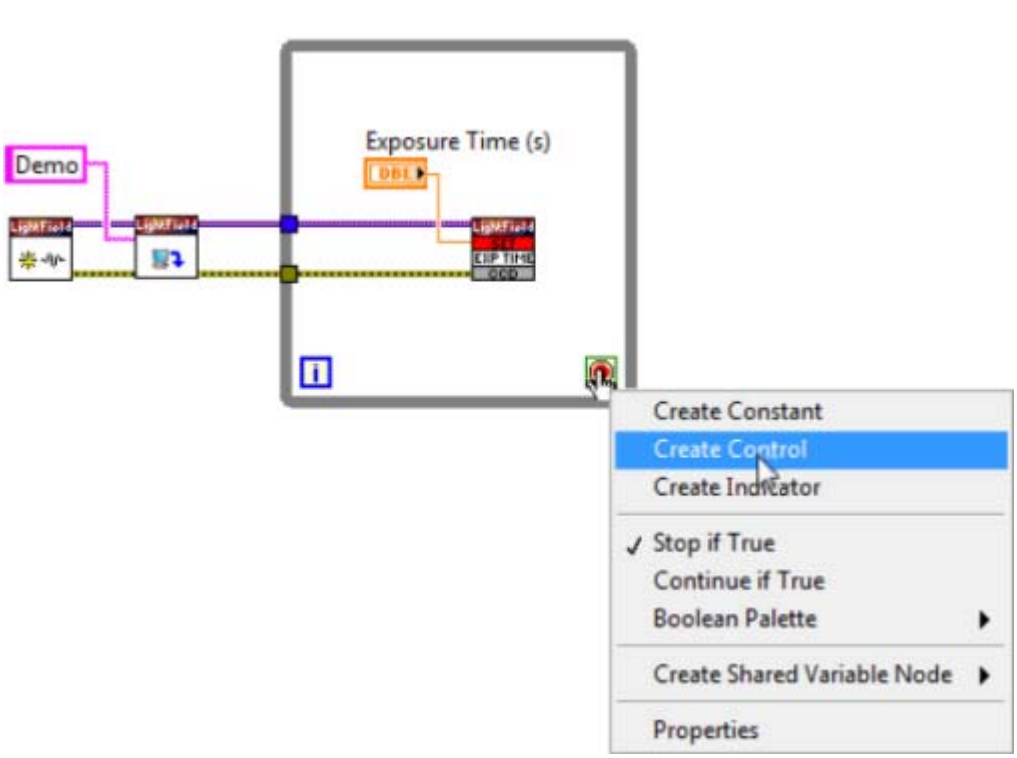
If desired, reposition the `Exposure Time (s)` control similar to that shown in Figure 3-17.

Figure 3-17: Finalizing the Set Exposure Time Control



24. Click the  button in the lower right-hand corner of the `While Loop` object and select **Create Control** from the pop-up menu displayed. See Figure 3-18.

Figure 3-18: Create a Stop Automation Application Control



4411-0150_018

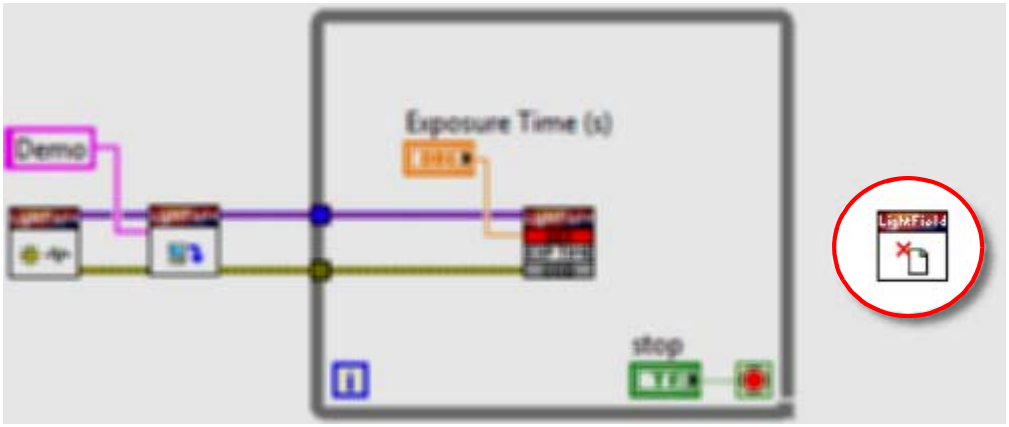
A Stop control has now been added, similar to that shown in Figure 3-19

Figure 3-19: Typical Stop Control

4411-0150_019

25. Within the **Methods** folder in the **Project Explorer**, locate the `LFA_Close.vi` method, and drag it into the **Block Diagram**. See Figure 3-20.

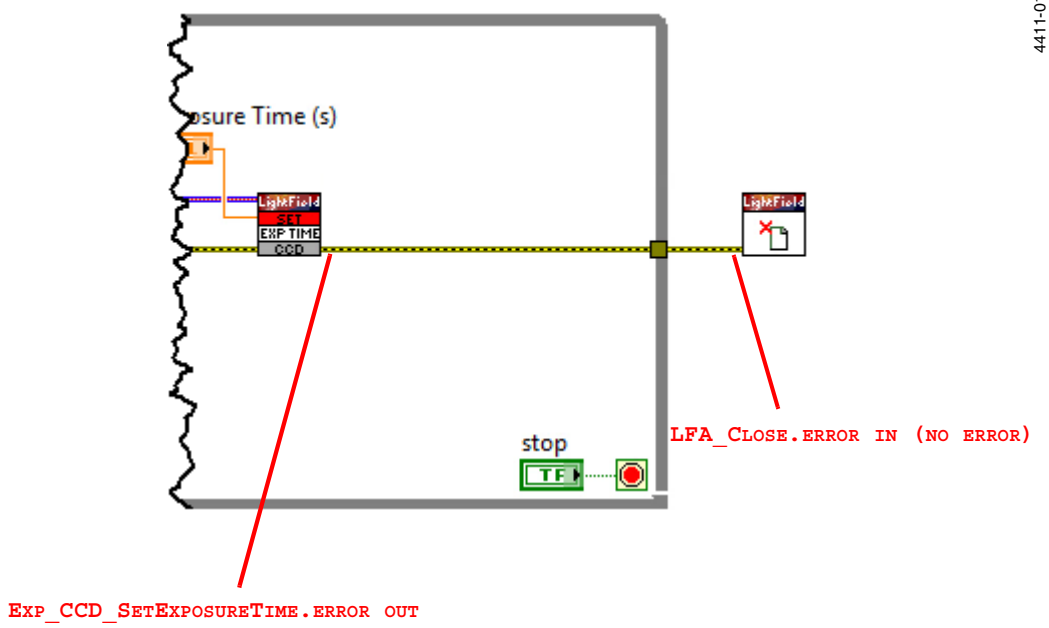
Figure 3-20: Adding `LFA_Close.vi` Method



4411-0150_020

26. On the **Block Diagram**, connect `Exp_CCD_SetExposureTime.error out` to `LFA_Close.error in (no error)`. See Figure 3-21.

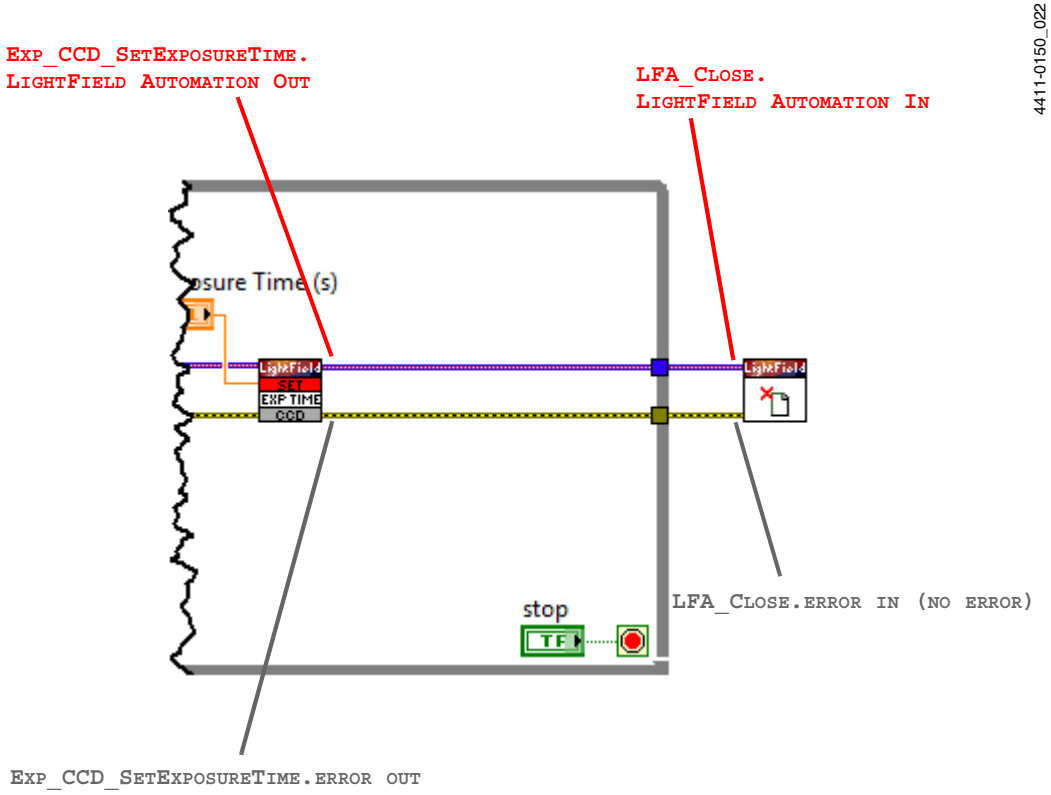
Figure 3-21: LFA_Close.vi Error Control Connections



4411-0150_021

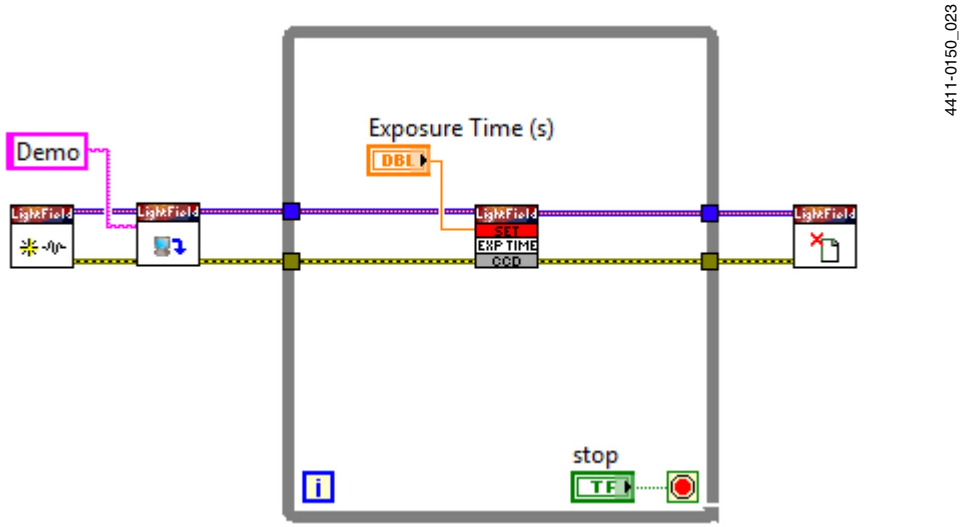
27. On the **Block Diagram**, connect `Exp_CCD_SetExposureTime.LightField Automation Out` to `LFA_Close.LightField Automation In`. See Figure 3-22.

Figure 3-22: Set Exposure Time Automation Control Connections



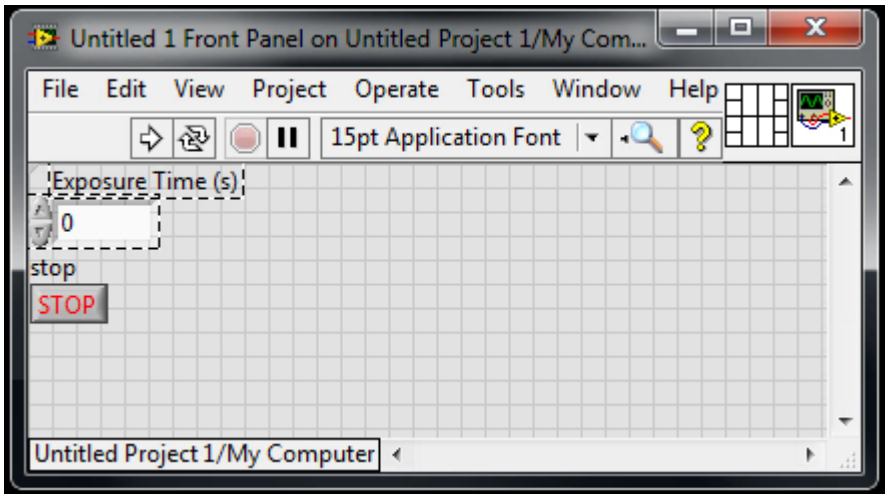
28. Figure 3-23 illustrates the block diagram of the completed Automation application.

Figure 3-23: Complete Automation Application Block Diagram



29. If desired, minimize the **Block Diagram** and **Project Explorer** windows, leaving the **Front Panel** as the only active window. The **Front Panel** may be resized if desired. See Figure 3-24.

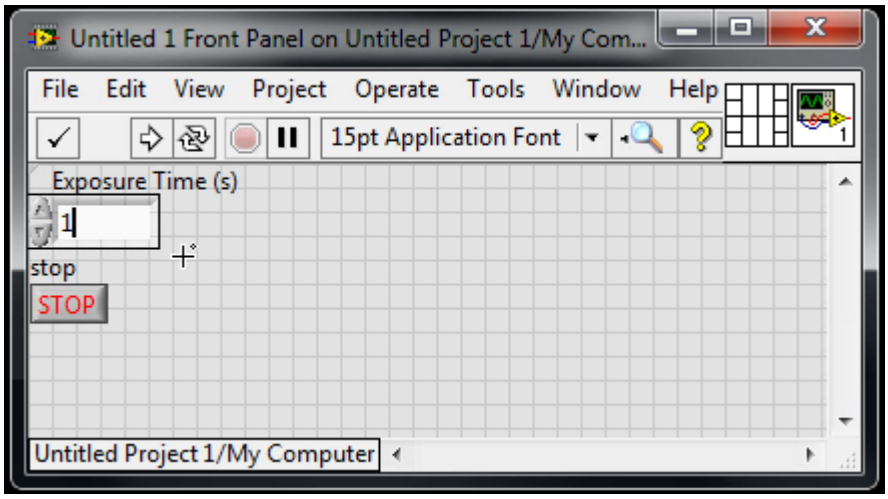
Figure 3-24: Front Panel for Exposure Time Automation Application



4411-0150_024

30. Within the **Exposure Time (s)** field, enter the desired exposure time, in seconds (e.g., 1 second,) as shown in Figure 3-25.

Figure 3-25: Exposure Time Configured for 1 Second



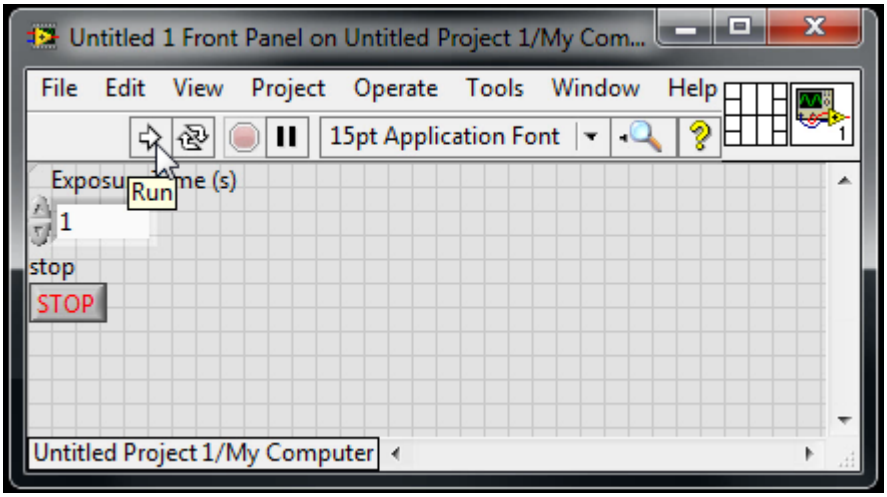
4411-0150_025

NOTE:

The ✓ indicates that the value entered is valid.

31. Click **Run**  to launch the newly created application. See Figure 3-26.

Figure 3-26: Running an Automation Application



4411-0150_026

Once launched, the application will:

- Launch and initialize LightField with an empty workspace;
- Load the previously saved experiment named **Demo** using the configuration settings saved with the experiment.

NOTE:

Depending on the size of the LightField desktop and where it is positioned on the monitor, resize and reposition it on the monitor for ease of accessibility and viewing of the LabVIEW **Front Panel**.

32. If necessary, expand the **Common Acquisition Settings** expander to view the **Exposure Time** configuration setting.

33. Within the **Front Panel**, manually enter/configure a new **Exposure Time** in the data field provided. Notice that the corresponding **Exposure Time** value within LightField is updated essentially in real-time.

NOTE:

If an invalid value is attempted to be entered, LabVIEW will display an error message pop-up window.

34. When done, click **STOP**  which will:

- Disconnect all hardware, and
- Close LightField.

35. Once LightField has closed, the LabVIEW application will halt execution.

This page is intentionally blank.

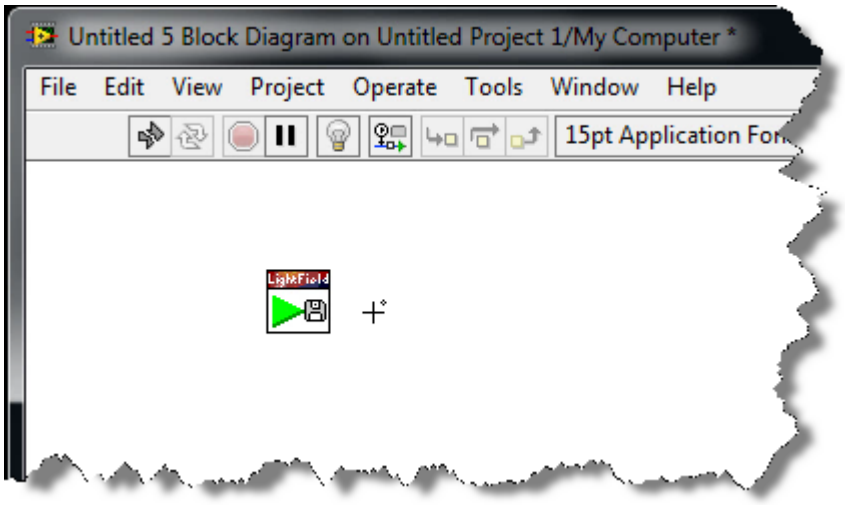
Chapter 4: Data Structures

This chapter briefly describes the data structures used when data are brought into LabVIEW from LightField.

Perform the following procedure to display LightField data structure information in LabVIEW:

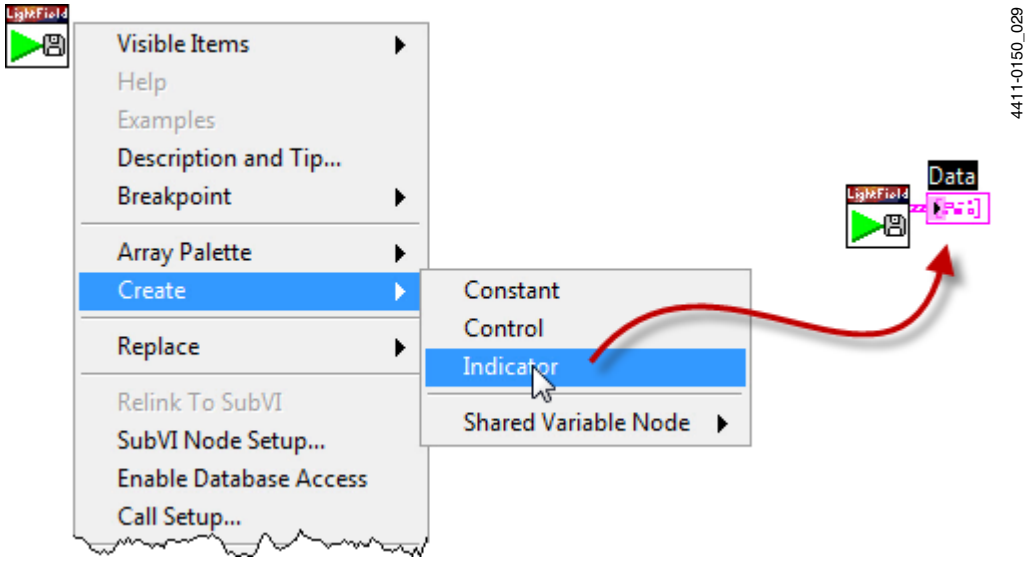
1. Within the LabVIEW **Project Explorer**, locate the `LFA_AcquireWithAutosave.vi` method, and drag it into the **Block Diagram**. See Figure 4-1.

Figure 4-1: LFA_AcquireWithAutosave.vi Method



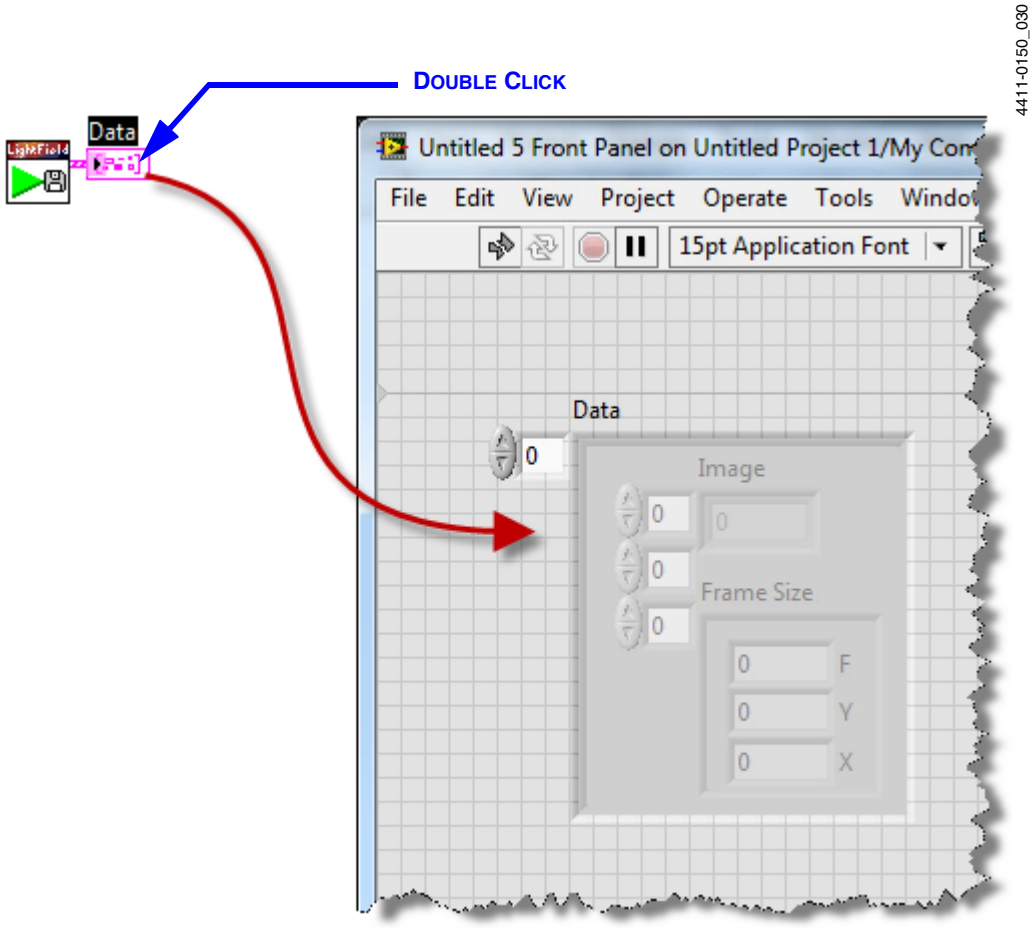
2. Right-click on the `LFA_AcquireWithAutosave.vi` method and select **Create –> Indicator** from the pop-up menu displayed. This will create a **Data** wire coming out of the `LFA_AcquireWithAutosave.vi` method containing the data as shown in Figure 4-2.

Figure 4-2: Create Data Wire from LFA_AcquireWithAutosave.vi Method



3. Double click on the **Data** wire to display the **Complex Data Structure** that comprises the data within the LabVIEW Front Panel. See Figure 4-3.

Figure 4-3: Complex Data Structure



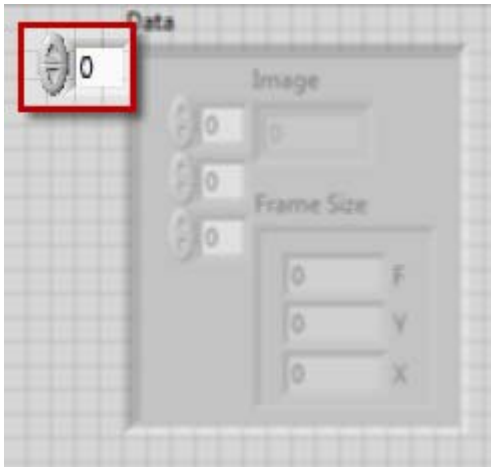
4.1 Data Structure Elements

The **Complex Data Structure** is actually an array of **Clusters** where each element of the array corresponds to an individual Region of Interest (ROI) from the camera. For example:

- Array Element 1 is ROI 1
- Array Element 2 is ROI 2

The **Cluster** index is located in the upper left-hand corner of the **Data Structure** object. See Figure 4-4.

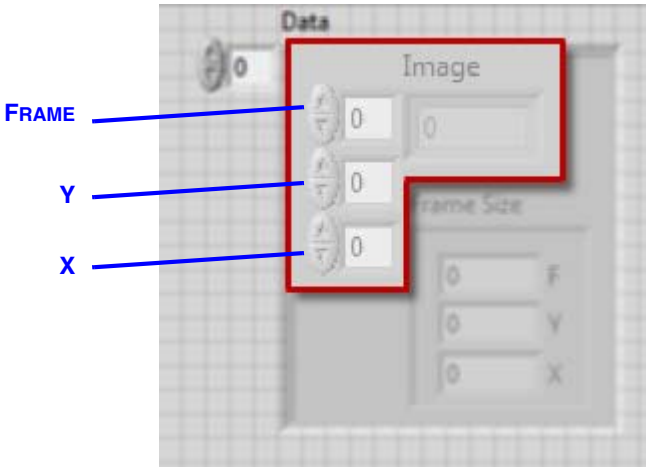
Figure 4-4: Cluster Index



4411-0150_031

Each **Cluster** is comprised of a three-dimensional array in which actual Image intensity data corresponding to individual pixels is stored. The indices for Image Intensity data, as well as the corresponding Intensity value for the specified pixel, are located within the **Image** section highlighted in Figure 4-5.

Figure 4-5: Image Array Indices and Intensity Data



4411-0150_032

The three array indices, illustrated in Figure 4-5, are:

- Frame Number

  Indicates the total number of frames that have been acquired.

  For example:

  — When one frame has been acquired, F = 1.

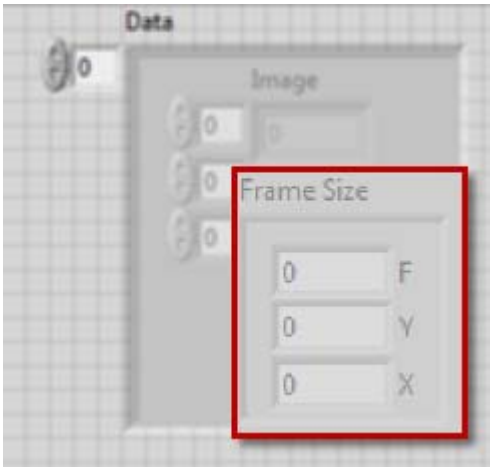
  — When 37 frames of data have been acquired, F = 37.
- Y

  Specifies the row number within the array.
- X

  Specifies the column number within the array.

Finally, the dimensions of each Frame are supplied in the **Frame Size** region highlighted in Figure 4-6.

Figure 4-6: .Frame Size Data



The dimension parameters within this section are:

- Frame Number

  Indicates the total number of frames that have been acquired.

  For example:

  — When one frame has been acquired, F = 1.

  — When 37 frames of data have been acquired, F = 37.
- Y

  Specifies the number of rows in the specified frame.
- X

  Specifies the number of columns in the specified frame.

The dimension information is used to determine the size of the incoming image.

Chapter 5: Parse Data Object Method

One of the most useful LabVIEW methods for use with LightField is the `Parse Data Object Simple.vi` method. Data being received from LightField will be either:

- A single spectrum, or
- A single image.

The `Parse Data Object Simple.vi` accepts a single LightField data input, parses the data, and returns the following two data outputs:

- ROI 1, Frame 1

  This is image data.
- ROI 1, Frame 1, Row 1

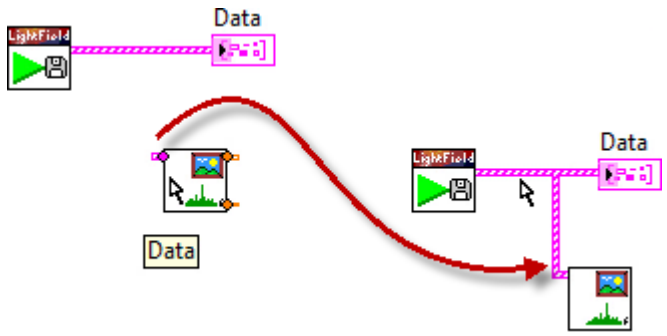
  This is spectroscopic data.

Implementing the `Parse Data Object Simple.vi` method simplifies application design by not requiring the use/implementation of index array and cluster VIs when accessing data.

Perform the following procedure to implement the `Parse Data Object Simple.vi` method:

1. Create a new `LFA_AcquireWithAutosave.vi` method as described in Chapter 4, Data Structures, on page 25, step 1 and step 2.
2. Within the **Project Explorer**, expand the **Data Slice** folder, and drag the `Parse Data Object Simple.vi` method onto the **Block Diagram**.
3. Connect the **Data** port on the `Parse Data Object Simple.vi` method to the Data object associated with the `LFA_AcquireWithAutosave.vi` method as illustrated in Figure 5-1.

Figure 5-1: Connecting Parse Data Object Simple Method



4411-0150_035

4. Left-click on each of the two data outputs on the `Parse Data Object Simple.vi` and select **Create –> Indicator** to create the two parsed data outputs:
   - `ROI 1, Frame 1`
   - `ROI 1, Frame 1, Row 1`

   See Figure 5-2.

Figure 5-2: Creating Parsed Outputs