

Distributing Applications with the LabVIEW Application Builder

Publish Date: Aug 23, 2013

Overview



(<http://zone.ni.com/devzone/cda/tut/p/id/7117>) This article is part of a series on software engineering practices and tools for large application development in LabVIEW.

Click here (<http://zone.ni.com/devzone/cda/tut/p/id/7117>) to view the list of other articles

Software developers rarely intend for a finished application to be used only on the development computer. To address this issue, National Instruments provides the LabVIEW Application Builder, a user-friendly tool that enables LabVIEW users to configure build specifications:

- **Stand-alone applications** that can be distributed to users who have the LabVIEW Run-time Engine.
- **Packed project libraries** that package multiple files in one location with a project library as the root file.
- **Shared libraries** that call LabVIEW code from text-based programming languages, such as LabWindows/CVI, Microsoft Visual C++, and Microsoft Visual Basic.
- **Source distributions** that package LabVIEW source files, which can be distributed to other LabVIEW developers.
- **Installers** that install stand-alone applications and shared libraries created for the Windows platform.
- **Zip files** that are useful for distributing a large amount of code as a single, portable file.

Using build specifications differs significantly depending on the version of LabVIEW you use. Refer to the *LabVIEW Help* for the most detailed and up-to-date information about the Application Builder. You can access the *LabVIEW Help* by selecting **Help»LabVIEW Help** in LabVIEW.

The LabVIEW Professional Development System includes the Application Builder. If you use the LabVIEW Base Package or Full Development System, you can purchase the Application Builder separately.

Note: You must have LabVIEW 8 or higher to complete these tutorials. Some text may differ in the software depending on which version of LabVIEW you use.

Table of Contents

1. Preparing the LabVIEW Project
2. Stand-Alone Applications
3. Packed Project Libraries
4. Shared Libraries
5. Installers (Windows)
6. Source Distributions
7. Zip Files

Building Applications with the Application Builder

The LabVIEW Application Builder leverages the organization provided by the LabVIEW Project, which organizes and manages all of the files associated with an application. These files include VIs, project libraries, documentation, data files, hardware configuration and more. The Application Builder creates applications from user-specified files in a LabVIEW project, and individual build settings are saved in the project.

The following examples use existing VIs to step through the tasks for configuring build specifications in LabVIEW. Download the attached VIs to complete each example. You must have a project open and saved to configure a build specification.

1. Preparing the LabVIEW Project

1. Download and unzip the file `tutorial_files.zip` attached to this document.
2. Select **File»New Project** to create a project.
3. Right-click **My Computer** and select **Add»Folder** from the shortcut menu to add a directory of files to the project as a folder. Navigate to the files from `tutorial_files.zip`.
4. Click the **Current Folder** button to add the directory to the project. The **Application Builder** folder appears under **My Computer** in the **Project Explorer** window.
5. Save the project as `TutorialProject` in a new directory named `AB Tutorial`. Use this project to complete the examples.

Note If you previously hid **Build Specifications** in the **Project Explorer** window, you must display the item again to access it. To view **Build Specifications**, click the **Filter Project View** button and select **Build Specifications**.

2. Stand-Alone Applications

You can develop stand-alone applications (executables) that only require the LabVIEW Run-Time Engine to run on systems without LabVIEW installed. Complete the following steps in each task to configure the application build specification and build and run the application. Refer to the *LabVIEW Help* for the most detailed and up-to-date information about building stand-alone applications. You can access the *LabVIEW Help* by selecting **Help»LabVIEW Help** in LabVIEW.

Configuring the Build Specification

1. From the **Project Explorer** window, right-click **Build Specifications** and select **New»Application** from the shortcut menu to display the **Application Properties** dialog box.
2. Complete the following steps on the **Information** page of the dialog box.
 1. Enter `Tutorial-EXE` in the **Build specification name** text box. The text box contains the name of the application build specification that appears under **Build Specifications** in the **Project Explorer** window.
 2. Enter `Tutorial.exe` in the **Target filename** text box. The text box contains the filename on disk for the executable file.
 3. Click the **Browse** button next to the **Destination directory** text box and navigate to the `AB Tutorial` directory. Create a subdirectory named `App`. Double-click the `App` directory and click the **Current Folder** button to select the directory. When you build the application, the executable file appears in this directory.
3. From the **Category** list on the left, click **Source Files** to display the **Source Files** page, which contains settings you can use to specify which files are included and how they are included in the application. Complete the following steps on the dialog box page.
 1. From the **Project Files** tree, click the **Tutorial Files** folder to expand it. Select **AB Tutorial GUI.vi**. Click the **Add Item** arrow button next to the **Startup VIs** listbox to add the VI to the listbox. VIs you add to the **Startup VIs** listbox open and run when you run the application.
 2. From the **Project Files** tree, select **AB Tutorial MeanMedianMode.vi**. Click the **Add Item** arrow button next to the **Always Included** listbox to add the VI to the listbox. Files you add to the **Always Included** listbox are in the application, but the application calls them dynamically as needed.
 3. Repeat the previous step to add the following files to the **Always Included** listbox: `AB Tutorial StndDev.vi`, `AB Tutorial Histogram.vi`, and `datafile.txt`. Do not add the other files, which are for use in other examples.

Note If you move a VI to the **Always Included** listbox on the **Source Files** page, but you choose a destination in the **Source File Settings** page that is not the same destination as the built application, the Application Builder moves all dependencies of the VI to the destination you designate, rather than keep the dependencies in the built application. If two or more VIs that are always included or top-level VIs call a VI and try to move it to two different locations, the Application Builder moves the VI and all subVIs to the built application. To make a VI that is specified as **Include if referenced** move to a new location, you must specify the VI as **Always Included** on the **Source Files** page.

4. From the **Category** list, select the other dialog box pages to explore the settings you can configure in a build specification for an application. For this exercise, do not change the settings on these dialog box pages.
5. From the **Category** list, click **Preview** to display the **Preview** page. Click the **Generate Preview** button to view a preview of the files that the Application Builder creates when you build the application. You can use this preview to verify that the Application Builder will create the files you want in the directories you want. You can correct errors before you build the application.
6. Click the **OK** button to close the dialog box and update the project with the build specification settings. The **Tutorial-EXE** build specification appears under **Build Specifications** in the **Project Explorer** window.

Building and Running the Application

1. Save the project. LabVIEW does not save build specifications settings on disk until you save the project.
2. Right-click the **Tutorial-EXE** build specification and select **Build** from the shortcut menu to build the application. A dialog box displays the progress of the build. When the dialog box displays the **Build Complete** message, click the **Done** button.
3. Navigate to the `AB Tutorial\App` directory. The directory should contain the executable file `Tutorial.exe`, the LabVIEW configuration settings file `Tutorial.ini`, and the `data` directory, which contains the support files `datafile.txt` and `lvnllys.dll`.
4. Double-click `Tutorial.exe` to run the application. A file dialog box prompts you to navigate to and select the `datafile.txt` file, located in the `data` directory created with the application.
5. Click the **Acquire Data** button to load data in the buffer. Click the **Graph Data** button to display the data in the graph on the right. Select an analysis to perform in the **Type of Analysis** section and click the **Analyze** button to display statistics on the data.
6. Click the **Stop** button to close the application.

3. Packed Project Libraries

Complete the following steps in each task to configure the build specification settings and build the packed project library.

Refer to the *LabVIEW Help* for the most detailed and up-to-date information about building shared libraries. You can access the *LabVIEW Help* by selecting **Help»LabVIEW Help** in LabVIEW.

Configuring the Build Specification

1. From the **Project Explorer** window, right-click **Build Specifications** and select **New»Packed Library** from the shortcut menu to display the **Packed Library Properties** dialog box.
2. Complete the following steps on the **Information** page of the dialog box.
 1. Enter `Tutorial-packedlib` in the **Build specification name** text box. The text box contains the name of the shared library build specification that appears under **Build Specifications** in the **Project Explorer** window.
 2. Enter `Tutorial-packedlib.lvlibp` in the **Target filename** text box. The text box contains the filename on disk for the executable file.
 3. Click the **Browse** button next to the **Destination directory** text box and navigate to the `AB Tutorial` directory. Create a subdirectory named `PackedLib`. Double-click the `PackedLib` directory and click the **Current Folder** button to select it. When you build the shared library, the files appear in this directory.
3. From the **Category** list on the left, click **Source Files** to display the **Source Files** page, which contains settings you can use to specify which files are included and how they are included in the shared library. Complete the following steps on the **Source Files** page of the dialog box.
 1. From the **Project Files** tree, click the **Tutorial Files** folder to expand it. Select `AB Tutorial Library.lvlib`. Click the **Add Item** arrow button next to the **Top-level Library** listbox to add the VI to the listbox.
 2. From the **Project Files** tree, select `AB Tutorial MeanMedianMode.vi`. Click the **Add Item** arrow button next to the **Always Included** listbox to add the VI to the listbox. Files you add to the **Always Included** listbox are in the packed library, but the packed library calls them dynamically as needed.
 3. Repeat the previous step to add the `datafile.txt` file to the **Always Included** listbox.
4. From the **Category** list, select the other dialog box pages to explore the settings you can configure in a build specification for a packed library. For this exercise, do not change the settings on these dialog box pages.
5. From the **Category** list, click **Preview** to display the **Preview** page. Click the **Generate Preview** button to view a preview of the files that the Application Builder creates when you build the packed library. You can use this preview to verify that the Application Builder will create the files you want in the directories you want. You can correct errors before you build the packed library.
6. Click the **OK** button to close the dialog box and update the project with the build specification settings. The **Tutorial-packedlib** build specification appears under **Build Specifications** in the **Project Explorer** window.

Building the Packed Library

1. Save the project. LabVIEW does not save build specifications settings on disk until you save the project.
2. Right-click the **Tutorial-packedlib** build specification and select **Build** from the shortcut menu to build the shared library. A dialog box displays the progress of the build. When the dialog box displays that the build is complete, click the **Done** button.
3. Navigate to the `AB Tutorial\PackedLib` directory. The directory should contain the packed library file `Tutorial-packedlib.lvlibp` and the support files `datafile.txt` and `lvanlys.dll`.

4. Shared Libraries

Complete the following steps in each task to configure the shared library build specification and build the shared library. Refer to the *LabVIEW Help* for the most detailed and up-to-date information about building shared libraries. You can access the *LabVIEW Help* by selecting **Help»LabVIEW Help** in LabVIEW.

Configuring the Build Specification

1. From the **Project Explorer** window, right-click **Build Specifications** and select **New»Shared Library** from the shortcut menu to display the Shared Library Properties dialog box.
2. Complete the following steps on the **Information** page of the dialog box.
 1. Enter `Tutorial-sharedlib` in the **Build specification name** text box. The text box contains the name of the shared library build specification that appears under **Build Specifications** in the **Project Explorer** window.
 2. (Windows) Enter `Tutorial-sharedlib.dll` in the **Target filename** text box. The text box contains the filename on disk for the executable file. (Mac OS X) Enter `Tutorial-sharedlib.framework`. (Linux) Enter `Tutorial-sharedlib.so`.
 3. Click the **Browse** button next to the **Destination directory** text box and navigate to the `AB Tutorial` directory. Create a subdirectory named `SharedLib`. Double-click the `SharedLib` directory and click the **Current Folder** button to select it. When you build the shared library, the files appear in this directory.
3. From the **Category** list on the left, click **Source Files** to display the Source Files page, which contains settings you can use to specify which files are included and how they are included in the shared library. Complete the following steps on the dialog box page.
 1. From the **Project Files** tree, click the **Tutorial Files** folder to expand it. Select **AB Tutorial MeanMedianMode.vi**. Click the **Add Item** arrow button next to the **Exported VIs** listbox to add the VI to the listbox and display the **Define VI Prototype** dialog box. LabVIEW exports VIs you add to the **Exported VIs** listbox so you can send and return values to and from the shared library.
 2. Complete the following steps from the **Define VI Prototype** dialog box.
 1. Click the **return value** parameter in the **Parameters** listbox. Verify that the **Function Prototype** description begins with `void`.
 2. Verify that the **Parameters** listbox includes an input named **Array** and outputs named **mean**, **median**, and **mode**.
 3. If the function does not contain these parameters, click the blue **Add Function (+)** button to add the parameters and set the values to input or output.
 4. Click the **OK** button to save the configuration settings and close the **Define VI Prototype** dialog box.
 3. From the **Project Files** tree, select **AB Tutorial StndDev.vi**. Click the **Add Item** arrow button next to the **Exported VIs** listbox to add the VI to the listbox and display the **Define VI Prototype** dialog box.
 4. From the **Define VI Prototype** dialog box, complete the steps listed above for the `AB Tutorial MeanMedianMode.vi`, but with different parameters. Verify that the **Parameters** listbox includes an input named **Array** and an output named **standardDeviation**.
 5. From the **Project Files** tree, select **AB Tutorial Histogram.vi**. Click the **Add Item** arrow button next to the **Exported VIs** listbox to add the VI to the listbox and display the **Define VI Prototype** dialog box.
 6. From the **Define VI Prototype** dialog box, complete the steps listed above for the `AB Tutorial MeanMedianMode.vi`, but with different parameters. Verify that the **Parameters** listbox includes an input named **Array** and an output named **HistogramOut**.
4. From the **Category** list, select the other dialog box pages to explore the settings you can configure in a build specification for a shared library. For this exercise, do not change the settings on these dialog box pages.
5. From the **Category** list, click **Preview** to display the **Preview** page. Click the **Generate Preview** button to view a preview of the files that the Application Builder creates when you build the shared library. You can use this preview to verify that the Application Builder will create the files you want in the directories you want. You can correct errors before you build the shared library.
6. Click the **OK** button to close the dialog box and update the project with the build specification settings. The **Tutorial-sharedlib** build specification appears under **Build Specifications** in the **Project Explorer** window.

Building the Shared Library

1. Save the project. LabVIEW does not save build specifications settings on disk until you save the project.
2. (Mac OS X) To build part of a shared library in LabVIEW, Application Builder must use gcc, the standard command-line compiler for Unix and Mac OS X operating systems. Use Xcode, the standard development environment for Apple computers, to install gcc. You can download Xcode for free from the Mac App Store.
3. Right-click the **Tutorial-sharedlib** build specification and select **Build** from the shortcut menu to build the shared library. A dialog box displays the progress of the build. When the dialog box displays the **Build Complete** message, click the **Done** button.
4. Navigate to the `AB Tutorial\SharedLib` directory. The directory should contain the shared library file `Tutorial-sharedlib.dll`, the file `Tutorial-sharedlib.h` that contains parameter prototypes for the shared library functions, the LabVIEW configuration settings file `Tutorial.ini`, and the `data` directory, which contains the support files for the shared library.

You can call the shared library from a number of programming languages, including LabVIEW and C.

5. Installers (Windows)

You can create a single installer that will install the LabVIEW Run-Time Engine, any hardware drivers, and MAX configuration settings necessary for running the application. Complete the following steps in each task to configure the installer build specification and build and run the installer. Refer to the *LabVIEW Help* for the most detailed and up-to-date information about building installers for Windows. You can access the *LabVIEW Help* by selecting **Help»LabVIEW Help** in LabVIEW.

Configuring the Build Specification

1. From the **Project Explorer** window, right-click **Build Specifications** and select **New»Installer** from the shortcut menu to display the **Installer Properties** dialog box.
2. Complete the following steps on the **Product Information** page of the dialog box.
 1. Enter `Tutorial-Installer` in the **Build specification name** text box. The text box contains the name of the installer build specification that appears under **Build Specifications** in the **Project Explorer** window.
 2. Enter `AB Tutorial` in the **Product name** text box. The text box contains the name that displays when the installer runs.

3. Click the **Browse** button next to the **Installer destination** text box and navigate to the `AB Tutorial` directory. Create a subdirectory named `Installer`. Double-click the `Installer` directory and click the **Current Folder** button to select the directory. When you build the installer, the files appear in this directory.
3. From the **Category** list on the left, click **Source Files** to display the **Source Files** page, which contains settings you can use to specify which files are included and how they are included in the installer.
4. From the **Project View** tree, click the **Tutorial-EXE** build specification to select it. From the **Destination View** tree, click the **TutorialProject** folder to select it. Click the **Add Item** arrow button next to the **Destination View** tree to add the application build specification files to the installer.
5. From the **Category** list, click **Shortcuts** to display the **Shortcuts** page, which you can use to create shortcuts to the files that the installer installs and to determine the location of the shortcuts. Complete the following steps on the **Shortcuts** page to create a shortcut on the desktop.
 1. Click the **Add Shortcut** button to display the **Select Target File** dialog box. Double-click the `Tutorial.exe` file to select it.
 2. Change the setting in the **Directory** ring to `[All Users Desktop]` so the shortcut appears on the desktop.
 3. Change the text in the **Name** text box to `Tutorial Application`.
 4. Delete the text in the **Subdirectory** text box so the shortcut appears directly on the desktop and not within a subdirectory.
6. From the **Category** list, click **Additional Installers** to display the **Additional Installers** page, which you can use to include National Instruments product installers that you need in the installer build. The **National Instruments Installers to Include** listbox contains a list of the installers available on the computer.
7. Place a checkmark in the **NI LabVIEW Run-Time Engine** checkbox, if a checkmark is not already present, to include the LabVIEW Run-Time Engine in the installer and all its components. Users who want to run the application must have the LabVIEW Run-Time Engine installed.
8. From the **Category** list, select the other dialog box pages to explore the settings you can configure in a build specification for an installer. For this exercise, do not change the settings on these dialog box pages.
9. Click the **OK** button to close the dialog box and update the project with the build specification settings. The **Tutorial-Installer** build specification appears under **Build Specifications** in the **Project Explorer** window.

Building and Running the Installer

1. Save the project. LabVIEW does not save build specifications settings on disk until you save the project.
2. Right-click the **Tutorial-Installer** build specification and select **Build** from the shortcut menu to build the installer. A dialog box displays the progress of the build. When the dialog box displays the **Installer build complete** message, click the **Done** button.
3. Navigate to the `AB Tutorial\Installer\Volume` directory. The directory should contain subdirectories named `bin`, `license`, and `supportfiles`, and the files `nidist.id`, `Setup.exe`, and `setup.ini`.
4. If you want to run the installer to test the LabVIEW Run-Time Engine installation, copy the files to a computer that does not include LabVIEW and run `Setup.exe`.

6. Source Distributions

Complete the following steps in each task to configure the source distribution build specification and build the source distribution. Refer to the *LabVIEW Help* for the most detailed and up-to-date information about building source distributions. You can access the *LabVIEW Help* by selecting **Help»LabVIEW Help** in LabVIEW.

Configuring the Build Specification

1. From the **Project Explorer** window, right-click **Build Specifications** and select **New»Source Distribution** from the shortcut menu to display the **Source Distribution Properties** dialog box.
2. Complete the following steps on the **Information** page of the dialog box.
 1. Enter `Tutorial-Dist` in the **Build specification name** text box. The text box contains the name of the source distribution build specification that appears under **Build Specifications** in the **Project Explorer** window.
 2. Click the **Browse** button next to the **Destination directory** text box and navigate to the `AB Tutorial` directory. Create a subdirectory named `Distribution`. Double-click the `Distribution` directory and click the **Current Folder** button to select the directory. When you build the source distribution, the files appear in this directory.
3. From the **Category** list on the left, click **Source Files**, which contains settings you can use to specify which files are included and how they are included in the source distribution. Complete the following steps on the dialog box page.
 1. From the **Project Files** tree, click the **Tutorial Files** folder to expand it.
 2. Select all of the items in the folder.
 3. Click the **Add Item** button next to the **Always Included** listbox to add the items to the listbox.
4. From the **Category** list, select the other dialog box pages to explore the settings you can configure in a build specification for a source distribution. For this exercise, do not change the settings on these dialog box pages.
5. From the **Category** list, click **Preview** to display the **Preview** page. Click the **Generate Preview** button to view a preview of the files that LabVIEW creates when you build the source distribution. You can use this preview to verify that LabVIEW will create the files you want in the directories you want. You can correct errors before you build the source distribution.
6. Click the **OK** button to close the dialog box and update the project with the build specification settings. The **Tutorial-Dist** build specification appears under **Build Specifications** in the **Project Explorer** window.

Building the Source Distribution

1. Save the project. LabVIEW does not save build specifications settings on disk until you save the project.
2. Right-click the **Tutorial-Dist** build specification and select **Build** from the shortcut menu to build the source distribution. A dialog box displays the progress of the build. When the dialog box displays the **Build Complete** message, click the **Done** button.
3. Navigate to the `AB Tutorial\Distribution` directory. The directory should contain the VIs and support files from the project as well as other internal files that LabVIEW determined the source distribution needed to include.

7. Zip Files

Complete the following steps in each task to configure the zip file build specification and build the zip file. Refer to the *LabVIEW Help* for the most up-to-date information about building zip files. You can access the *LabVIEW Help* by selecting **Help»LabVIEW Help** in LabVIEW.

Configuring the Build Specification

1. From the **Project Explorer** window, right-click **Build Specifications** and select **New»Zip File** from the shortcut menu to display the **Zip File Properties** dialog box.
2. Complete the following steps on the **Zip Information** page of the dialog box.
 1. Enter `Tutorial-Zip` in the **Build specification name** text box. The text box contains the name of the zip file build specification that appears under **Build Specifications** in the **Project Explorer** window.

2. Click the **Browse** button next to the **Destination** text box and navigate to the `AB Tutorial` directory. Create a subdirectory named `Zip`. Double-click the `Zip` directory. Enter `Tutorial.zip` in the file name field and click the **Select** button. When you build the zip file, a file with the name you specified appears in the directory you specified.
3. From the **Category** list on the left, click **Source Files** to display the Source Files page, which contains settings for the files to include in the zip file.
4. From the **Source Files** page, place a checkmark in the **Zip entire project** checkbox so the zip file includes all the VIs and support files in the project. This is a faster option than moving files from the **Project Files** tree to the **Included Items** listbox if you want to include all the project items in the zip file.
5. From the **Category** list, select the Zip File Structure page other dialog box pages to specify the file structure for a zip file. For this exercise, do not change the settings on this page.
6. Click **Preview** to display the Preview page. Click the **Generate Preview** button to view a preview of the compressed file structure that the Application Builder creates when you build the zip file. You can use this preview to verify that the Application Builder will include the files you want from the directories you want. You can correct errors before you build the zip file. Notice that the zip file includes all project items and the project file (`.lvproj`) itself.
7. Click the **OK** button to close the dialog box and update the project with the build specification settings. The **Tutorial-Zip** build specification appears under **Build Specifications** in the **Project Explorer** window.

Building the Zip File

1. Save the project. LabVIEW does not save build specification settings on disk until you save the project.
2. Right-click the **Tutorial-Zip** build specification and select **Build** from the shortcut menu to build the zip file. A dialog box displays the progress of the build. When the dialog box displays the **Build complete** message, click the **Done** button.
3. Navigate to the `AB Tutorial\Zip` directory. The directory should contain the file `Tutorial.zip`.

You can use a standard decompression tool to decompress the zip file and view the VIs and support files it includes.

Customer Reviews

23 Reviews | Submit your review (http://sine.ni.com/apps/utf8/nidz_display_comments.create_comment?p_title=Distributing+Applications+with+the+LabVIEW+Application+Builder&p_doc_id=DZTUT-3303-EN&p_doc_type=DZTUT&p_url=http://www.ni.com/tutorial/3303/en&p_doc_url_id=3303)