FAQ

Frequently Asked Questions

LabVIEW™ Application Builder

This addendum supplements the information in the *LabVIEW Application Builder Release Notes*. For the latest information, check National Instruments Web Support at `http://www.natinst.com/support/lvsupp.htm`.

Changes in the Application Build Process

The process of building applications is simplified. It is no longer necessary to build an intermediate library (`.llb`) using the **Save With Options** dialog box. With version 5.1 of the Application Builder, you can create applications directly from your source VIs. *You no longer need to perform the* **Save With Options/Application Distribution** *step*.

New Run-Time Engine Architecture

On Windows NT/98/95, you must install the LabVIEW Run-Time Engine on computers that run your LabVIEW applications. In previous versions, this engine was built into the executable. It is now a separate component. On non-Windows platforms, the run-time engine is included in the executable file.

Where can I find the LabVIEW Run-Time Engine installer?

If you purchased the Application Builder separately, you can find the run-time engine installer on your Application Builder CD in the `redist\RunTime` directory.

If you have the LabVIEW Professional Development System (PDS), the run-time engine installer is automatically copied to the `LabVIEW\Applibs\installs\RunTime` directory on your hard drive when you install LabVIEW.

You also can download the run-time engine installer from the National Instruments web site at `http://www.natinst.com/labview/sample_over.htm` from the Drivers & Updates link.

How do I distribute the run-time engine installer along with my own installer?

If you create an installer for your application, by default LabVIEW copies the run-time engine installer from LabVIEW\AppLibs\installs\RunTime to the RunTime directory in the directory with your setup.exe. When you execute your setup.exe, your installer automatically calls the run-time engine installer. If it does not find the correct directory, it prompts the user to insert the first disk of the run-time engine installer.

I want to distribute my installer on 1.44 MB floppy disks. How do I copy the run-time engine installer to floppy disks?

The run-time engine installer requires two floppies. From the run-time engine installer directory, copy data2.cab to the second floppy and all other files to the first floppy.

What additional steps do I need to perform if I am using 3D Graphs, Report Generation, or DataSocket?

If you are using these features and you are building an installer, copy the appropriate installer for each component to your LabVIEW\Applibs\installs directory. By default, these installers are not installed with LabVIEW. They are located on your LabVIEW CD in the Redist directory. For example, if you use the 3D Graph, copy the files from *X*:\redist\3Dgraph, where *X* is the letter of your CD-ROM drive, to LabVIEW\Applibs\installs\3Dgraph.

Then, click the **Advanced…** button in the **Application Builder Installer** tab and check the feature you are using in the **Advanced Installer Settings** dialog box. The Application Builder then copies the installers for these components from LabVIEW\Applibs\installs to the location of your installer.

Can I simply copy lvrt.dll into the same directory as my executable, rather than installing the run-time engine?

When the run-time engine is installed, lvrt.dll (the engine DLL) is copied to Program Files\National Instruments\shared\LabVIEW Run-Time*Version#* by default. Copying this DLL into the same directory as your executable will suffice, because the executable looks in its directory when searching for the engine DLL. You lose some modularity with this method, because each executable needs a duplicate of lvrt.dll in its directory. If you use this option, include lvrt.dll as a support file in your build.

How do I enable the web server in an application?

You must distribute your executable, its preferences (.ini) file, the lvjpeg.dll, and the lvpng.dll to the target computer. The .ini file should be in the same directory as the application. On Windows platforms, the DLLs must be in the same directory as lvrt.dll (usually Program Files\National Instruments\Shared\LabVIEW Run-Time). On non-Windows platforms, the DLLs should be in the same directory as the application.

Are there any special considerations if my program uses ActiveX events or containers?

Two files, `ax-events.dll` and `ole_lv5container.dll`, provide support for ActiveX events and containers. If your application includes ActiveX event VIs, `ax-events.dll` is automatically included as a support file. `ole_lv5container.dll` is installed when the run-time engine is installed. If you are not using the run-time engine installer, you must manually include `ole_lv5container.dll` (found in `LabVIEW\Resource`) as a support file and place it in the same directory as the executable.

How do I build an executable containing MATLAB or HiQ script nodes?

Create a `Script` subdirectory in the directory with your executable. If you are using the MATLAB script node, copy `matscript.dll` from `LabVIEW\Resource\Script` into your `Script` directory. If you are using the HiQ script node, copy the `hiqscript.dll` into your `Script` directory. You can perform these steps as part of the application build by adding these DLLs as support files with a custom destination (place them in a `Script` subdirectory in the same directory of your executable) using the **Custom Destinations** button in the **Source Files** tab.

How do I include a custom About screen with my application?

Write a VI with a front panel that describes your application. The name of the VI must start with `About`. Include this VI as a dynamic VI when you build the application. This enables an **About** option in the **Help** menu of your executable to execute your About VI.

Why doesn't my application use the custom icon I specified?

LabVIEW can import black and white and color icons in each of two resolutions, 16×16 and 32×32, for a total of four possible icons. When the icon file does not contain all four icons, LabVIEW uses its default icon for the missing icon. You can avoid this by using your icon editor to specify all icon types.

What should I do if I receive errors while building the application?

First, ensure you saved the VIs you are building in LabVIEW 5.1. You can mass compile your VIs for this purpose. Second, do not use the **Save With Options/Application Distribution** process to build an intermediate library for the application build.

NATIONAL INSTRUMENTS™

322513A-01   Jun99