Programmatically Build Applications in LabVIEW

Publish Date: aug 01, 2011 | 8 Ratings | **2.75** out of 5 | Print | 3 Customer Reviews | Submit your review

Overview

This article is part of a series on software engineering practices and tools for large application development in LabVIEW.

Click here to view the list of other articles

This example shows how to batch build several distributions in LabVIEW using a programmatic build process. If you are using LabVIEW 2011 or later, use the Build VI to programmatically build applications.

1. Introduction

Many applications require frequent compilation into an executable (`.exe`) or a dynamic link library (`. dll`). An application might require building the source code into an executable every night or every ten minutes. Prior to LabVIEW 8.0 the application builder required the creation of a unique *.bld file to correspond to every configuration your application required. You could then call that file from a batch file.

When the LabVIEW project was introduced in LabVIEW 8.0, build specifications were integrated into the project. To provide the previous capability of batch compilation, LabVIEW now provides a VI called BuildTargetBuildSpecification.vi that programmatically builds a distribution using a predefined build specification. This example shows how to use that feature to batch build applications from the command line. This enables you to build applications from a task automation program such as Windows Task Scheduler.

2. Details of the example

Back to Top

The example `programmaticbuild.vi` shows how to programmatically build several projects using command line arguments. Using the BuildTargetBuildSpecification.vi (which is packaged with the LabVIEW Application Builder), this example simply iterates through the command line arguments and uses BuildTargetBuildSpecification.vi as a subVI to build projects according to their specified build specifications. Any LabVIEW version 8.0 or later with Application Builder installed has this function available in `<LabVIEW directory>\vi.lib\AppBuilder.`

To run this example, open a command line window. Browse to the LabVIEW directory, and enter the following line in the command line window:

```
labview.exe "<Location of programmaticbuild.vi>" -- "<Project 1>.lvproj" "<Project 2>.lvproj" "<Project 3>.lvproj" <…> "<Project n>.lvproj"
```

This example is configured to run automatically when the VI opens so that the projects can be built from the command line window. You can implement this process into any batch processing system that is set up (such as Windows Task Scheduler) by configuring the batch program to call labview.exe with the previous arguments.

This example include a built-in error logger that writes the file `error_log.txt` to the temporary directory specified in the **Options** dialog box. This log displays information about the build process. Select **Tools»Options** to display the **Options** dialog box, and select **Paths** from the **Category** list to configure the temporary directory. You can edit the build process used in this example. Additionally, the example contains a sequence structure that closes LabVIEW when the VI finishes executing, depending on the value of the Boolean constant.

Back to Top

Customer Reviews

3 Reviews | Submit your review

No feedback during build - 27-jul-2009

By Dennis Jackson, Sawtooth Labs.

A couple of issues: 1. I made the same update the other two comments here made; NI should fix the posted VI to correct this error for all users. 2. The program gives no feedback during the build process -- nothing is output to a log file (preferably) or the screen. 3. Once the build starts, you can't stop it even with the Stop button on programmiticbuild2.vi. (on Windows XP).

Write text file fails in programmicbuild2.vi - 13-mrt-2009

By Mikkel Bie, Deductor ApS.

There is a programming mistake in this VI: The close file operation needs to be moved outside the for-loop to make the write of status to textfile work.

Small Problem - 15-jan-2009

I ran into a small problem using this VI. It closes the file reference in between each iteration of the command line argument loop. I fixed this problem by moving the file close to outside of the loop. After the fix, it became a useful utility. I am running a Windows XP machine with LabVIEW 8.5. It may operate differently on another system.