

Application Note: BitFlow Application Development in Borland C++ Builder™ 6.0

Introduction

The BitFlow Standard Development Kit (SDK) is implemented with platform neutrality in mind. However, compiling and executing an application on any setup other than that originally designed on will likely pose some challenges, and such is the case with Borland C++ Builder™ (BCB) 6. This Application Note demonstrates how to configure BCB for use with the BitFlow SDK, and in particular, how to get the included BitFlow examples compiling and running.

The Essentials

Included with this Application Note should be two folders, “Include” and “Lib.” “Include” supplies the file “BF_in_BCB.cpp” and the corresponding “BF_in_BCB.h,” which can be included in BCB source files to provide a compatibility layer for a few functions used by the BitFlow SDK and SDK examples that BCB does not support; additional compatibility modifications may be necessary to run and compile some of the SDK examples. “Lib” supplies the BitFlow SDK library files in the OMF format used by BCB (the library files shipped with the SDK by default are in the COFF format, required for the Microsoft tools).

In order for programs using the BitFlow SDK to compile correctly in BCB, the BCB compiler must be set in *Force C++ mode*, as described in the **Configuring a Project** section below.

The entirety of the following section provides step-by-step instructions for compiling and running a basic application using the BitFlow SDK in BCB.

Note that there are two types of examples included with the BitFlow SDK, console applications (i.e. ones that run in a command window) and applications with a full Windows GUI. The BitFlow GUI examples are built on top of Microsoft's MFC libraries which are not compatible with BCB. This Application Note will only discuss working with console based examples.

Getting Started and a Simple Example

Before any development can begin, one must have acquired and installed a copy of Borland C++ Builder™ 6 and the BitFlow SDK, the latter of which is available for download at the [BitFlow.com Downloads page](http://www.bitflow.com/Downloads).

Upon installation or first run of BCB 6, if using Windows Vista or Windows 7, one may be prompted to enable a compatibility mode, a suggestion which should be ignored. On these versions of the operating system, make sure that compatibility mode is disabled for BCB by right clicking in the windows menu on *All Programs > Borland C++ Builder 6 > C++ Builder 6* and selecting the *Properties* item in the context menu that opens. Navigate to the *Compatibility* tab and make sure that the *Run this program in compatibility mode for* check-box is deselected, clicking the *Apply* button if any change is made.

Preparing a Development Folder

Using Windows Explorer, or whatever file manager may be preferred, create a folder for development such as “C:\Users\[Current User]\Documents\BCB Projects\.” Within this new folder, copy and paste the provided

“Include” and “Lib” folders, making sure to include all of their contents.

At this time, any existing project files and folders should be copied and pasted into the development folder. For the purpose of the current example, the “C:\BitFlow SDK 5.30\Examples\BiSimple\” folder – or the corresponding folder in whichever version of the BitFlow SDK that is installed – and all of its contents should be copied (make sure that the copied files are not write protected).

Configuring a Project

Opening BCB 6, one is presented with an empty default project, which the user should be closed. Select from the main program menu *File > New > Other*, and in *New* tab of the *New Items* dialog that will open, select the *Console Wizard* option and click the *OK* button. The *Console Wizard* dialog will open, in which the desired *Source Type* should be selected, and, if any code already exists, the *Specify project source* check-box should be selected and the corresponding main-file location entered into the text field. Any such code should by now be located in the development folder created before, and for this example should be at “C:\Users\[Current User]\Documents\BCB Projects\BiSimple\CircularSimple.c.” The *OK* button can now be selected.

Select from the main menu, *View > Project Manager* to open the *Project Manager* window. Right click on the new *ProjectX.exe* item and select *Save*, continuing to save the project in the desired sub-folder of the development folder created before; save with the name “CircularSimple.bpr” in the “BiSimple” sub-folder for this example. Similarly, right click on the *ProjectGroupX* item and select *Save Project Group*, saving into the development folder itself; use the name “BitFlow_Examples.bpg” for this example.

In the *Project Manager* window, right click on the just renamed project, selecting the *Options...* item. In the *Project Options for [Project Name].exe* window that opens, navigate to the *C++* tab and select the *Force C++ Compile* check-box. In the *Directories/Conditionals* tab, add to the *Include path* listing under *Directories* the path of the Include folder located in the development folder; “C:\Users\[Current User]\Documents\BCB Projects\Include\” for the current example. As well, add to the *Include path* listing “C:\BitFlow SDK 5.30\Include\,” or the corresponding folder in whatever version of the BitFlow SDK is installed. Select the *OK* button.

Again, right click on the current project in the *Project Manager* window, selecting this time the *Add...* item. Navigate to the “Include” folder within the development folder, selecting the “BF_in_BCB.cpp” file and selecting the *OK* button (make sure the *Files of type* drop-down menu is set to “*.cpp”). In the same manner as before, open the *Add to project* dialog again, this time navigating to the “Lib” folder within the development folder and adding any of the BitFlow SDK libraries listed that the project will need (make sure the *Files of type* drop-down menu is set to “*.lib”); for the present example, the only library files needed are “BFD.lib,” “BFer.lib,” “Bid.lib,” “Cid.lib,” and “DispSurf.lib.” Additional library files can be added at any time, and all of the files may be added if it is not known which are needed.

Now, open in the *Editor* window any and all C and/or C++ project files that require access to the BitFlow SDK functions, adding to them the line

```
#include <BF_in_BCB.h>
```

which should generally appear prior to any other includes and code in the given file. The “CircularSimple.c” file will be the only file needing modification in this example. This step may not be necessary in every case, and an experienced software developer should be able to determine the appropriateness for the given project. Save any modified files.

Finally, select from the main menu *Project > Build [Project Name]* to build the current project and select *OK* to dismiss the *Compiling* window after compilation has completed. If no errors are produced, select from the main menu *Run > Run* to execute the project.