



BaciBeans: A NetBeans Plugin for Concurrent Programs



Ashley Broadwell, Mentor: J. Stanley Warford

Division of Natural Science, Pepperdine University, Malibu, CA 90263

The Southern California Conferences
for Undergraduate Research
Whittier College, November 23, 2013

Abstract:

C-- is a programming language developed by M. Ben-Ari to teach principles of concurrent and distributed programming. It is a small subset of the C language but with the added features of concurrency, semaphores, and monitors. BACI is an acronym for Ben-Ari Concurrency Interpreter. JavaBACI is an open-source command line concurrency simulator used by students to compile and run C-- programs. NetBeans is an integrated development environment (IDE) that many students use in our courses to develop Java and C++ programs. The benefits of an IDE over a command line tool include an integrated text editor and the point-and-click convenience of a graphical user interface. This poster describes the development of a NetBeans plug-in we call BaciBeans that integrates JavaBACI into NetBeans, combining the familiarity and ease of use of NetBeans with the concurrency interpreter. The plugin allows students to write, compile, and run C-- programs within the familiar NetBeans IDE. To create the plugin we incorporated the JavaBACI runtime code, created a new project of type C--, registered a new file type with a ".cm" file extension for source files, created new build and run actions to invoke the corresponding JavaBACI commands, redirected the JavaBACI output to the NetBeans terminal, and registered file extensions for object files. Whereas before students used a separate text editor to create and save source files and issued command line calls to use the BACI interpreter, they now simply download the plugin and code just as they would a Java or C++ program in NetBeans. In the future we plan to add more functionality to the plugin, including an integrated option to use the debugger, syntax coloring and formatting, and code completion.

Background:

The book *Principles of Distributed and Concurrent Programming* [1], describes the C-- programming language to teach principles of concurrent and distributed programming. Tracy Camp's website BACI Systems [2], provides an open source Java executable and source code, JavaBACI, a simulator for executing C-- programs. NetBeans is an IDE that uses the plugin architecture based on the Visitor Design Pattern [3]. The Visitor Design Pattern is used to add functionality to an existing framework by subclassing its classes, without having to recompile the framework. BaciBeans provides the functionality and point-and-click convenience of an integrated development environment that allows students to write, compile, and run C-- programs within the familiar NetBeans IDE. This plugin offers advantages over the old JavaBACI practice of using a text editor to write code and use the command-line to build, compile, and run a C-- program. The NetBeans plugin is easy to install within the NetBeans IDE and there is no hassle with .profile setup and command line calls. The NetBeans plugin allows multiple project management along with C++ and Java programs in the same project group, allowing students to build, run, and debug all at the touch of a button, and to save, create, and open new projects with a .cm file and project folder. These are only a few of the advantages this plugin offers. The next section shows the user experience on the command line without BaciBeans. The following sections show the new functionality with BaciBeans and describe implementation details.

Without BaciBeans:

```
Ashley-Broadwells-MacBook-Pro:Dropbox AshleyBroadwells$ cd bacibeans/TestProject/
Ashley-Broadwells-MacBook-Pro:TestProject AshleyBroadwells$ vim TestProject.cm
Ashley-Broadwells-MacBook-Pro:TestProject AshleyBroadwells$ cat TestProject.cm
// File: TestProject.cm
// Author: Ashley Broadwell
// Date: Fall 2013

void main() {
    cout << "Hello World\n";
}

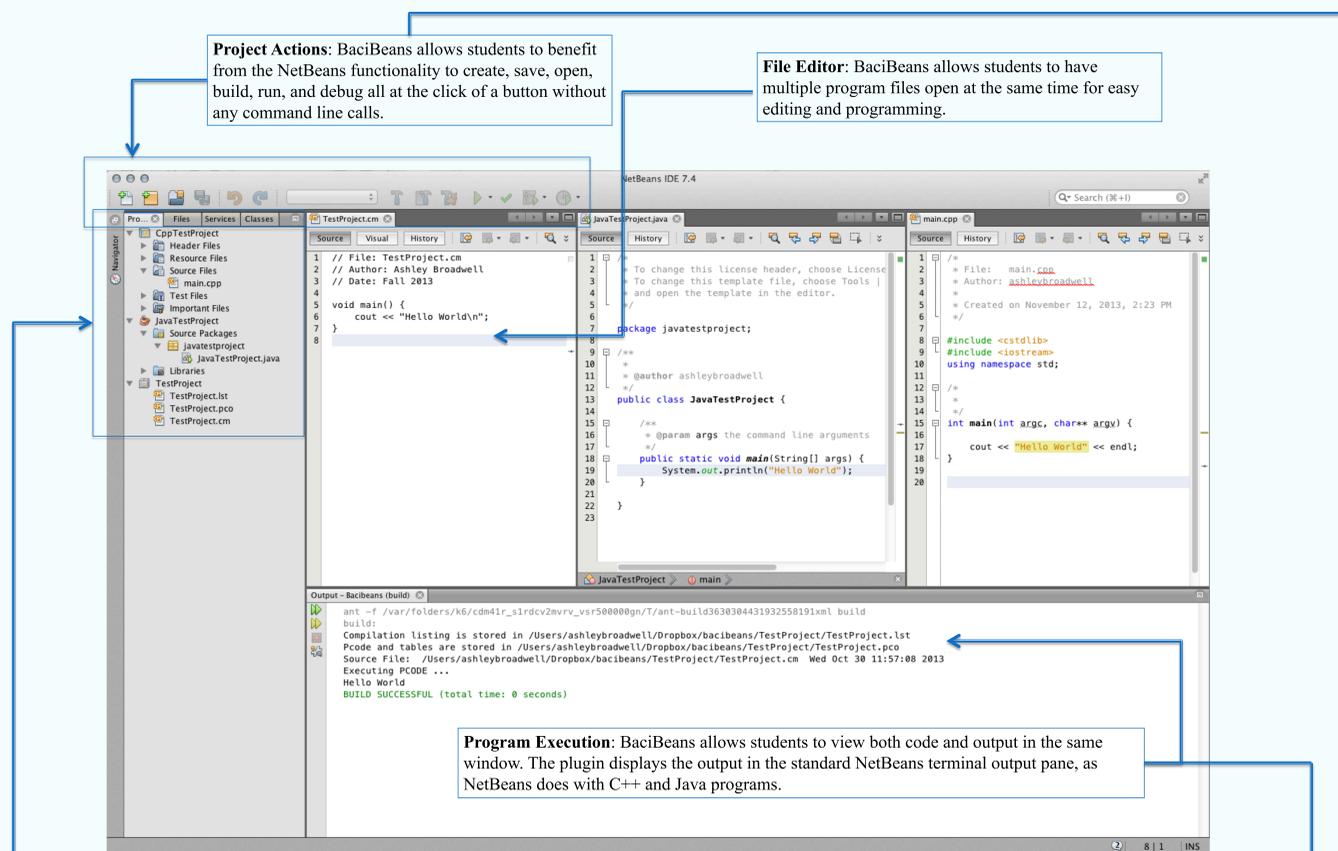
Ashley-Broadwells-MacBook-Pro:TestProject AshleyBroadwells$ javabaci bacc TestProject.cm
Compilation listing is stored in TestProject.lst
Pcode and tables are stored in TestProject.pco
Ashley-Broadwells-MacBook-Pro:TestProject AshleyBroadwells$ javabaci bainterp TestProject.pco
Source File: TestProject.cm Wed Oct 30 11:57:08 2013
Executing PCODE ...
Hello World
Ashley-Broadwells-MacBook-Pro:TestProject AshleyBroadwells$
```

Fig. 1 - Command line demonstration of creating, building, and running a C-- program.

Without the NetBeans Plugin students are required to edit, build, run, and debug all with the command line interface. They must download the JavaBACI executable and set the path in their profiles. After the initial setup they navigate to a folder, use a text editor such as vim and create their C-- files. After creating the files, to simulate concurrent execution they must call the JavaBACI build command to compile the .cm C-- source files and generate the .pco object files and .lst listing files. To debug they must input the debug statement on the command line requiring them to remember the debug flag. Finally, in order to run they must input the JavaBACI run command with the names of their files with the extension .pco to run their programs. Virtually all of these details are abstracted in the NetBeans plugin, making programming and C-- an easier experience. It's all at the click of a button.

With BaciBeans:

With the NetBeans plugin, C-- is easy to install and use. BaciBeans follows the simple installation protocol for NetBeans plugins. Separate installation of JavaBACI is not necessary as the JavaBACI executable code is built into the plugin. No longer are students forced to use the command line and memorize the build, debug and run commands and flags, they simply press a button in the toolbar of the IDE. The NetBeans plugin also allows for better project management. As illustrated in Figure 2, students can manage their C-- and Java projects together in the same project group. Further advantages are illustrated below.



Project Actions: BaciBeans allows students to benefit from the NetBeans functionality to create, save, open, build, run, and debug all at the click of a button without any command line calls.

File Editor: BaciBeans allows students to have multiple program files open at the same time for easy editing and programming.

Program Execution: BaciBeans allows students to view both code and output in the same window. The plugin displays the output in the standard NetBeans terminal output pane, as NetBeans does with C++ and Java programs.

Fig. 2 - The NetBeans IDE with the installed C-- plugin to allow for greater functionality and ease of programming in C-- including multi-project management.

Project Management: BaciBeans allows students to benefit from NetBeans project management, which allows C-- and Java projects to coexist in the same project group.

Implementation:

The BaciBeans plugin is a NetBeans module. It registers a new project type for C-- and the necessary file types to recognize .cm source files, .pco object files, and .lst listing files. When students create new projects, the standard NetBeans wizard appears. A BaciBeans package registers a new project type through a lookup function so that a C-- option appears in the NetBeans wizard. This new project type also creates a skeleton .cm source file. We customized icons and new project descriptions for the student to recognize and create C-- projects in their NetBeans IDE. We registered each action button with code to fetch the necessary file for the selected project along with the JavaBACI executable from the installed resource folder of the plugin. We created a run script with the necessary calls, flags, and files from the students' C-- project to perform the action. Instead of using an Ant script we wrote the action commands in a string, illustrated in Figure 3, and used a file writer to save the script and run it on the NetBeans command line outputting the results in the output pane.

```
File of = File.createTempFile("ant-build", ".cm");
BufferedWriter out = new BufferedWriter(new FileWriter(out.getAbsolutePath()));
out.write("ant -f "+out.getAbsolutePath()+" build\n");
out.close();
FileObject fof = FileUtil.toFileObject(out.getAbsolutePath());
ActionList.runTargetOf(fof, null);
FileUtil.refreshFromThisFile(fof);
} catch (IOException e) {
    System.out.println("IO error: "+ e);
}
}
```

Fig. 3 - The run script in the RunAction file.

BaciBeans allows students to easily create a new project by clicking on the existing NetBeans New Project toolbar button. Clicking on this button shows the C-- New Project option illustrated in Figure 4. BaciBeans generates a .cm source file when a new C-- project is created. Students can create a new .cm file or open an existing C-- project by clicking on the corresponding buttons on the existing NetBeans toolbar. If students want to open an existing C-- .cm file in NetBeans, they can place this file within a folder with the same name (without the file extension). BaciBeans automatically recognizes this configuration as a C-- project.

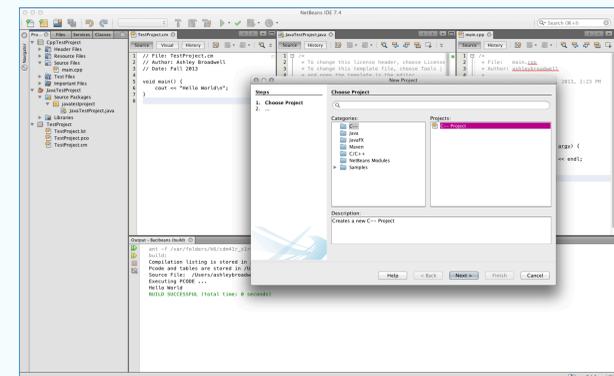


Fig. 4 - The new project wizard with the BaciBeans plugin.

Conclusions:

BaciBeans provides students with a new and improved way to program in C-- by allowing students to use the familiar NetBeans IDE. BaciBeans has several advantages over the existing system to edit, build, run, and debug on the command line interface using the JavaBACI [2] executable by incorporating the C-- project, file type, and actions within the NetBeans IDE. Students also benefit from the project management feature of NetBeans, which allows them to maintain Java, C++, and C-- programs coexisting in the same project group. BaciBeans is successfully used in the Programming Paradigms course at Pepperdine University. We have several plans for improving BaciBeans in the future, including C-- syntax highlighting and formatting and code completion capabilities.

Acknowledgements:

Support for this project was provided by the Natural Science Division at Pepperdine University and the Academic Year Undergraduate Research Initiative (AYURI) Fall 2013.

References:

[1] Ben-Ari, M. *Principles of Concurrent and Distributed Programming*, Addison-Wesley Pearson, Harlow, England, second edition, 2006.
[2] Camp, Tracy. *BACI System*. Colorado School of Mines. Web. 13 Nov. 2013. <http://inside.mines.edu/~tcamp/baci/baci.html>.
[3] Gamma, Erich, et al. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Longman, Inc. Reading, Massachusetts, 1994. Print.