CBCJVM: Applications of the Java Virtual Machine with Robotics: Setup, Help, and Tomorrow

Braden McDorman *(beta/catron)*
Benjamin Woodruff *(pipeep)*
Akshay Joshi *(axyjo)*
Jonathan Frias *(freakinjonathan)*

Norman Advanced Botball `braden@betabot.org`
Fletcher High School `odetopi.e@gmail.com`
American Community School of Abu Dhabi `me@akshayjoshi.com`
Robert E. Lee High `freakinjonathan@gmail.com`

CBCJVM: Applications of the Java Virtual Machine with Robotics: Setup, Help, and Tomorrow

1 Introduction

Welcome to the second CBCJVM paper! If you haven't already read "CBCJVM: Applications of the Java Virtual Machine with Robotics: Concept and Features," we would recommend doing so before reading this paper. In this paper we will look at the details of setting up CBCJVM on your operating system, as well as on your CBC. After that we will look at the room for expansion in CBCJVM's libraries and features, as well as how to get help with the features already present.

2 Setup

2.1 Background Information

It is strongly recommended you use CBCJVM on a POSIX compliant operating system such as Linux or Mac OS X. On Windows, everything should work, except for the few bash-based build scripts (the build process is mainly Ant and Maven based). Complete Windows support is a priority, and should be an easy project for anyone who wishes to take on the task.

Warning: Build instructions may change over time. Therefore, it is always a good idea to consult the CBCJVM wiki[4] for the latest information.

2.2 Ubuntu Setup

All of these commands, unless otherwise specified, are to be executed in a bash (terminal) environment. On a fresh Ubuntu install, that would be found under the top left "Applications" menu, "Accessories", "Terminal". If a command asks for your password, it is referring to your root (administrative) password. If you have no idea what this means, please contact your system administrator.

1. Open the "Software Sources" application by going to the "System" menu at the top left of your screen, "Administration", "Software Sources"

2. Ensure the box "Community-maintained Open Source software (universe)" is checked, if not, check it.

3. Close the window. If you are prompted to update your package list, click the "Reload" button.

4. Check and see if you have a copy of JDK 6 by running:

   ```
   $ javac -version
   ```

   You should see something like:

   ```
   $ javac 1.6.0_18
   ```

   If it gives a version prior to 1.6.x_x, update your system. If it complain that javac is not found, run

   ```
   $ sudo apt-get install default-jdk
   ```

5. Now, install the additional dependencies:

   ```
   $ sudo apt-get install ant git-core maven2
   ```

   If you already have one of those pieces of software, you can still run the command without fear. Aptitude won't attempt to reinstall the software.

6. Navigate into the directory you want the repository folder to be downloaded to, making a new one if necessary; Eg:

   ```
   $ mkdir "~/myFolder"
   $ cd "~/myFolder"
   ```

7. Pull (download) a copy of CBCJVM:

   - Download the latest version (possibly unstable):

     ```
     $ git clone git://github.com/catron/CBCJVM.git
     $ cd CBCJVM
     ```

- Download a specific ("stable") version:

  ```
  $ git clone git://github.com/catron/CBCJVM.git
  $ cd CBCJVM
  $ git tag -l
  $ git checkout 10.6
  ```

  (`git tag -l` is optional, it will list all available versions. Replace '10.6' with whatever version number you desire.)

If you are unsure as to what version you should download, it is recommended that you consult someone on the Botball community chat[1].

8. Compile CBCJVM:

   ```
   $ sh BuildEverything.sh
   ```

9. The CBC installer and corresponding README file can now be found in `CBCJVM/-installer/install`.

2.3 Mac OS X Setup

All of these commands, unless otherwise specified, are to be executed in a bash (terminal) environment. The default Mac OS X terminal can be found in your "Applications" folder, "Utilities", "Terminal". These steps should work on any version of OS X, but we can only verify them on on Snow Leopard (10.6).

1. If you haven't already[1], download and install the latest version of the Apple Developer Tools (XCode et al)[2] available for your system. Please note that you do not need the "iPhone" version, although it will work. It simply contains more packages and is larger than the "Mac" specific version.

2. If you haven't already, download and install MacPorts[3] to follow the steps provided there.

3. Install required tools for building CBCJVM

   ```
   $ sudo port install apache-ant git-core maven2
   ```

4. Navigate into the directory you want the repository folder to be downloaded to, making a new one if necessary; Eg:

   ```
   $ mkdir "~/myFolder"
   $ cd "~/myFolder"
   ```

[1] If you have to folder "Developer" in the ROOT of your drive, this probably means you already installed the Apple Developer Tools

[2] `http://developer.apple.com/technologies/xcode.html`

[3] `http://www.macports.org/install.php`

5. Pull (download) a copy of CBCJVM:

   - Download the latest version (possibly unstable):

     ```
     $ git clone git://github.com/catron/CBCJVM.git
     $ cd CBCJVM
     ```

   - Download a specific ("stable") version:

     ```
     $ git clone git://github.com/catron/CBCJVM.git
     $ cd CBCJVM
     $ git tag -l
     $ git checkout 10.6
     ```

     (`git tag -l` is optional, it will list all available versions. Replace '10.6' with whatever version number you desire.)

   If you are unsure as to what version you should download, it is recommended that you consult someone on the Botball community chat[4].

6. Compile CBCJVM:

   ```
   $ sh BuildEverything.sh
   ```

7. The CBC installer and corresponding README file can now be found in `CBCJVM/-installer/install`.

2.4 CBC Setup

After compiling CBCJVM on your computer, you must setup CBCJVM on all the CBCs you wish to use it on.

Warning: This will only work on a CBCv2. It will not work on a CBCv1, and it may or may not run on future versions of the CBC. Future versions of CBCJVM may or may not be compatible with different devices. Backwards compatibility is not guaranteed. **CBCJVM comes with no warranty**, and runs the slim possibility, as do all CBC mods, of turning your CBC into a very expensive paperweight. Neither the CBCJVM developers nor KIPR are responsible for you bricking your own device, and it is at KIPR's sole discretion whether or not to fix it. That said, CBCJVM avoids modifying certain portions of your CBC and attempts to remain as encapsulated as possible, making it improbable that it would damage your device.

2.4.1 Install

1. Copy the contents of the `CBCJVM/installer/install` directory to the ROOT of a flash drive.

2. Load `CBCJVMInstall.c` in the "Programs" section of the CBC (cbcui).

[4] `http://community.botball.org/irc`

3. Run.

2.4.2 Cleanup

1. Delete everything copied to the flash drive.

3 Ending Notes

3.1 Tomorrow

3.1.1 SemiSim

CBCJVM currently has a simulator, which is especially useful since you can use any JVM debugger of your choice. SemiSim would be a continuation of the Simulator, but all motor and servo commands (from `cbccore.low`) would be forwarded to a CBC over network. That way, you have the best of both worlds... You can debug your program using a full debugging tool, and execute it on the CBC seamlessly.

3.1.2 Google Voice Integration

Google Voice connectivity would allow a CBC to send and receive SMS (text) messages via WiFi, allowing for some interesting design ideas. This would most likely be accomplished with Google Voice Java[5]. This could be integrated into CBCJVM directly, allowing a background daemon to handle requests you register with the manager.

Imagine running your robot by simply sending a SMS message to it, or being able to quickly gather information from your CBC while it is still running. But that would only be the start. There could be all sort of unforeseen possibilities for this technology. This doesn't necessarily have to be done through Google Voice, it could be done as an IRC or Jabber (a free and open IM protocol), or better yet it could be done in a pluggable fashion, allowing people to use any web technology of their choice to communicate with it. Imagine a Twittering robot...

"Picked up the red tribbles! I'm going to head to the basket now!"

"The Nease iRobot Create robot is following me @FletcherHighLegoBot, why aren't you?"

3.2 Finding Documentation with JavaDoc

After running the build script, a series of HTML pages will be generated in `CBCJVM/cbc/-CBCJVM/doc`. You can open the file `index.html` from that directory with your favorite web-browser. This documentation is automatically generated from the CBCJVM source code, and is rebuilt with each execution of the build script.

For regular users of Java, this should look familiar. That's because official JavaSE/ME/-EE documentation is built using the same tool, JavaDoc. This generated documentation

[5] `http://code.google.com/p/google-voice-java/`

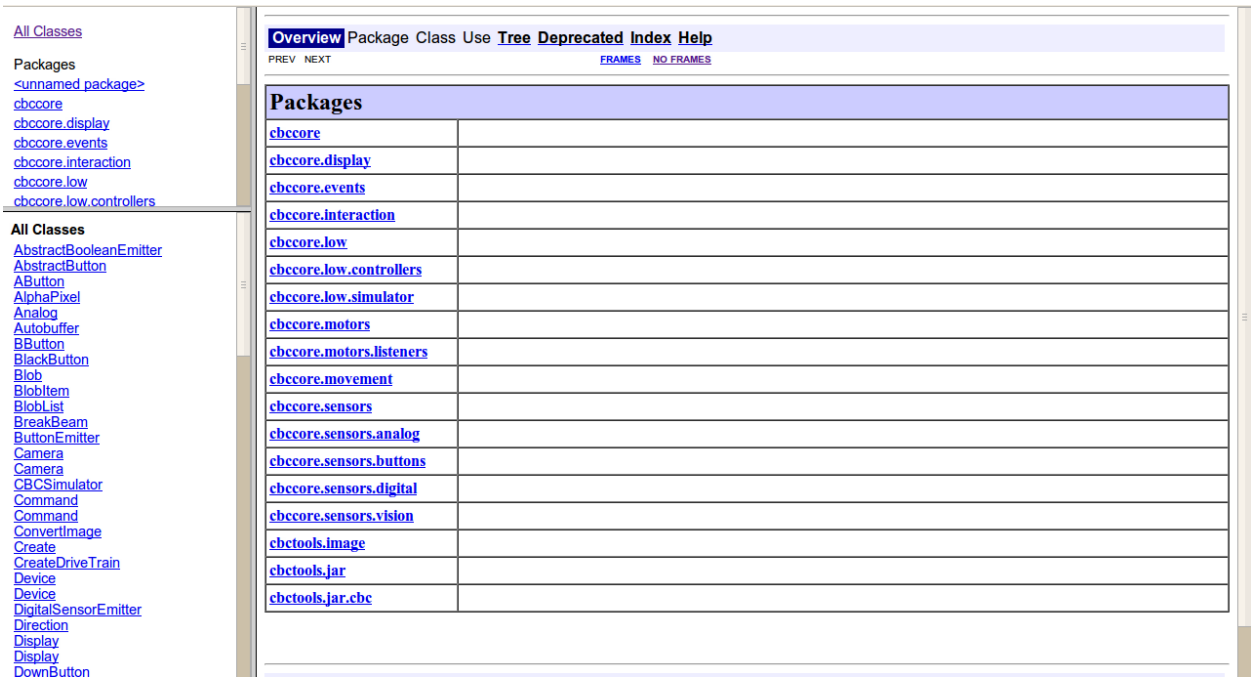

Figure 1: *The CBCJVM JavaDocs. The top left frame contains CBCJVM specific packages. The bottom left frame contains CBCJVM specific classes. The large frame is used to display information about the currently selected package or class.*

can be found on Sun's website[2]. All of the classes and methods found there should be available on the CBC, with the notable exception of AWT/Swing (the packages are actually there, but the CBC lacks a windowing server, making them useless), and a few other minor features not yet available in JamVM/GNU-Classpath (such as `System.nanoTime()`).

3.3 Finding Help on the Internet

This document cannot cover everything regarding CBCJVM or the technologies surrounding it. It only begins to scratch the surface. The Internet is an incredibly useful tool, but it pays to know where to look. This section attempts to address this, but above all, remember that Google is your friend.

3.3.1 General Programming and Java Specific Help

There exist a number of web sites that can help you locate general information on programming, theory and practice. A few of the best resources are listed here:

- `http://en.wikibooks.org/wiki/Java_Programming`
  Wikipedia also contains various programming articles, all of great quality. The are numerous Wikipedia articles on Programming Theory.

- `http://www.stackoverflow.com`
  A question-and-answer website for all types and kinds of programming. Search before asking a question. Registration *(free)* is required to ask questions.

3.3.2 CBCJVM and Botball Specific Help

- `http://community.botball.org`
  As if this one hasn't been plugged enough. Features a question-and-answer style forum. Registration *(free)* is required to ask questions.

- `irc.mintirc.net #botball` **or** `http://community.botball.org/irc`
  Pull up your favorite IRC client and join. Real-time community chat. Due to the limited number of CBCJVM users at the moment, people living in countries other than the US with different timezones may have trouble finding someone they can chat with. Registration *(free)* is not required, although suggested.

- `http://wiki.github.com/catron/CBCJVM`
  The official repository for all CBCJVM-specific information.

- `http://github.com/catron/CBCJVM/issues`
  If you find a bug in CBCJVM (which admittedly, in its early state, is not hard to do), or have a feature request, this is the place to report it/suggest it. Registration *(free)* is required to report questions.

4 License

CBCJVM is licensed under the GPLv3 [3]. This license requires that you distribute the source code of any modifications of CBCJVM along with any binary version of it that you distribute. We, the authors of CBCJVM, believe this constitutes a fair and just agreement. You are also permitted to use CBCJVM under any future version of the GPL that you wish.

5 Conclusion

We hope you're as excited as we are about CBCJVM, and we hope your team tries or uses the tools we have provided. Our project is growing quickly, so we hope to see you next year with an array of new features and enhancements!

References

[1] Botballer's Chat | Botball Community. `http://community.botball.org/botballers-chat`, 2010.

[2] Java Documentation — Oracle. `http://java.sun.com/javase/6/docs/api`, 2010.

[3] Richard Stallman et al. The GNU General Public License — GNU Project — Free Software Foundation (FSF). http://www.gnu.org/licenses/gpl-3.0.html, 06 2007.

[4] Benjamin Woodruff. Install/setup — cbcjvm — github. http://wiki.github.com/catron/CBCJVM/installsetup, 2010.