

100% Java, 100% Free

Howard Ong

Principal Consultant

Aurora Consulting Pty Ltd

Abstract

By traditional wisdom, good things don't come cheap. In recent years, Open-Source technology has seriously challenged this traditional notion. This paper introduces the audience to 3 powerful and yet freely downloadable Java development tools - NetBeans, CVS and Apache Ant. By explaining the key features of these tools and how they can be synergistically integrated into a powerful Java development environment, the author demonstrates the immense value-for-money of such an environment.

About The Author

A frequent presenter at Oracle Conferences, Howard has been an expert user of Oracle database and tools for the past 11 years and has been working on Internet Applications since 1995. Howard possesses in-depth experience in the planning and development of E-Business and other Web-Based Systems. Harboring a keen interest in the application of Oracle Technology in Internet application development, Howard and his team of consultants have helped many organisations deploy web-based solutions using technology such as the Oracle Portal, Oracle Toplink, Oracle Collaboration Suite, Oracle Files, Oracle Email, Oracle Database Server and third-party tools. Howard has worked with a wide variety of organisations such as government departments; and companies from mining, finance and transportation industries.

For more information on Howard or Aurora Consulting, visit <http://www.aurora-consult.com.au> or email info@aurora-consult.com.au.

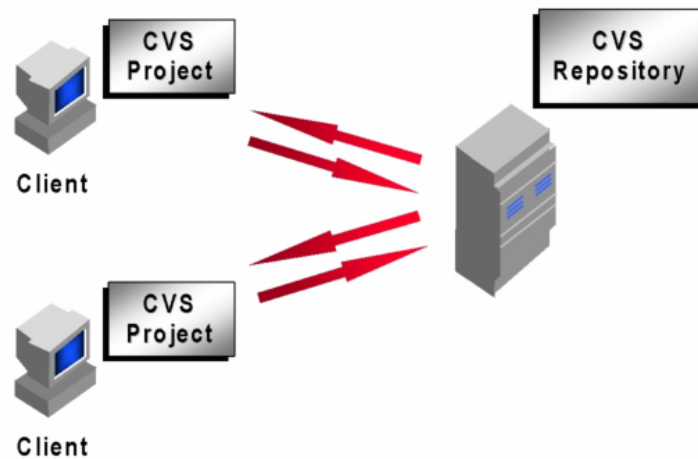
Introduction

As traditional wisdom goes : Good things are not cheap and cheap things are not good. For many years, this assertion rings true in the software industry. However, things are changing. Just as client-server and N-tier technology brought about the paradigm shift from large centralised servers to distributed computing, the advent of Open-Source software poses a serious challenge to the cheap-is-not-good paradigm.

Of all the inroads it made, the Open Source movement is none more dominant than in the sphere of Java software development. This paper examines and demonstrates 3 freely downloadable Java development tools and demonstrates how these can be integrated into a powerful Java development environment.

CVS

CVS (Concurrent Version Systems) is a highly popular open-source version control system. Its architecture consists of a central CVS repository, to which multiple CVS clients connect. These developer clients, each with a local CVS project, connect and synchronise to the central project on the CVS repository.



The CVS repository may reside on the local drive, a network drive, an intranet server or even an Internet server. The CVS client connects to the CVS repository via one of the following access method :

- `:local:`
CVS repository is on a file system accessible by the local client, which may be a local or network drive. No special server software is required.
- `:server:`
CVS repository resides on a remote server, accessible via a rsh (remote shell) program on the local client.
- `:ext:`
CVS repository resides on a remote server, accessible via a rsh program at the remote server.
- `:pserver:`
CVS repository resides on a remote server which requires password authentication. This is the most useful access method for Internet-based collaborative development.

The CVS repository is specified either with the `-d` parameter or the `cvsroot` environment variable, using the following syntax :

`[:method:][[:user][:password]@]hostname[:[:port]]path`

For example,

- `:local:c:\cvsrep`
- `:pserver:howard@cvsserver.aurora-consult.com.au:c:\cvsrep`

Natively, CVS supports a command line interface. In addition, open source GUI extensions to native CVS are abound. While it is not the intention of this paper to explain all commands supported by CVS, some of the more commonly used CVS commands are :

- `cvs import -m "log-message" project-name vendor release-tag`
Creates a new project in the CVS repository, from the content of the current directory.
- `cvs login`
Logs in to a :pserver: repository.
- `cvs checkout project-name`
Checks out a CVS project.
- `cvs add file-name`
Add the local file to the repository.
- `cvs update file-name`
Update the local file from the repository.
- `cvs commit -m "log-message" file-name`
Commit a local file changes to the repository.
- `cvs diff -r revision1 -r revision2 file-name`
Shows the difference between 2 revision of a file.
- `cvs logout`
Logs out from a :pserver: repository.

The popularity of CVS has led to the emergence of a number of related but independently developed products. Some of these are :

- CVSGui is a GUI front-end to CVS. CVSGui runs on popular GUI Operating Systems such as Windows, Macintosh and Linux. CVSGui is freely downloadable from <http://www.wincvs.org>.
- CVSNT is a CVS Server software developed for the Windows platform. CVSNT is freely downloadable from <http://www.cvsnt.com>.
- NetBeans is a Integrated Development Environment with CVS capability. NetBeans is explained later in this paper.

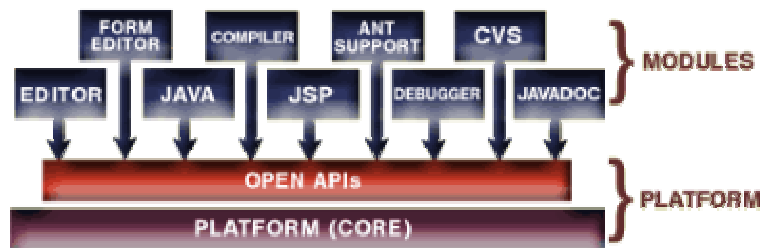
CVS may be downloaded from <http://www.cvshome.org>.

NetBeans

NetBeans is an Integrated Development Environment (IDE) for Java and other languages. It was originally developed by Sun Microsystems, who subsequently donated it to the Open-Source community. It is freely downloadable from <http://www.netbeans.org>.

Some of NetBeans key features include :

- NetBeans supports the plugging in of other software products to extend its core features. Some common plug-ins are :
 - Apache Ant, a Java build tool, described later in this paper.
 - CVS, the software version control tool, described earlier in this paper
 - Jakarta Tomcat, an application server for debugging servlets and JSPs. Tomcat is available from <http://jakarta.apache.org/tomcat>.
- Syntax highlighting for all supported languages. NetBeans supports the following languages :
 - Java
 - JSP
 - XML
 - HTML
 - C and C++
- Error checking code editor, supporting all the languages mentioned above.



Apache Ant

Apache Ant is a Java-based build tool. Written in Java, Ant is cross-platform and may be extended using Java classes. Each task is run by a java class. Ant's XML-based build files construct individual tasks into target trees. At runtime, Ant calls a target tree whereby various tasks are executed.

Stands for Another Neat Tool, Ant was originally part of the Tomcat code base, when it was donated to the Apache Software Foundation. Initially Ant was there to build Tomcat, and nothing else. Subsequently, other Open Source projects realised the usefulness of Ant and started using it as well. In January 2000, Ant was became an independent Apache project.

Although a thorough discussion of Ant syntax is beyond the scope of this paper, a number of common Ant tags are explained below :

- `<project name="project-name" default="default-destination-dir" basedir="base-dir"> ... </project>`
This is the top level tag.
- `<target name="target-name" depends="other-targets" description="description"> ... </target>`
Defines a collection of tasks to be performed. The execution of a target tree may be dependent on other target trees having executed first.
- `<property name="property-name" value="property-value"/>`
Sets a property by name and value. Once set, a property may be referred in other parts of the build file using the syntax `"${property-name}"`.
- `<javac srcdir="source-dir" destdir="destination-dir" classpath="class-path"/>`
Compiles all java files under the source directory, using the specified class path, and storing the compiled class files in the destination directory.

- `<jar destfile="jar-file" basedir="jar-dir"/>`
Jars all files in the jar directory into a jar file.
- `<copy file="source-file" tofile="destination-file"/>`
or
`<copy file="source file" todir="destination-file"/>`
Copies the file.
- `<fileset dir="directory"/>`
A type sub-tag supported by many of the task, a file set tag specifies one or more files to the task.

Ant may be downloaded from <http://ant.apache.org>.

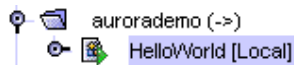
Bringing Them Together

Once integrated together, NetBeans, CVS and Ant forms a formidable combination.

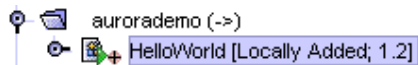
NetBeans & CVS

From the GUI environment of NetBeans the developer is able access all the version control functionality of CVS. For example,

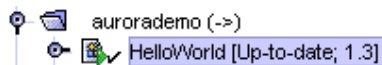
- File just created. CVS is unaware of this file and NetBeans marks it with a [Local] tag.



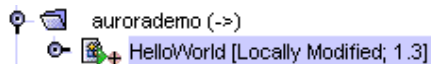
- Added to CVS. CVS is now aware of the file but a copy has yet to be uploaded to the CVS repository. NetBeans marks it with a + icon and the [Locally Added] tag.



- Change committed to CVS. The local copy of the file is in synch with the CVS repository copy. NetBeans marks it with a tick.



- Changes made to the local file. The local copy of the file is in now out of synch with the CVS repository copy. NetBeans marks it with a + icon and the [Locally Modified] tag.



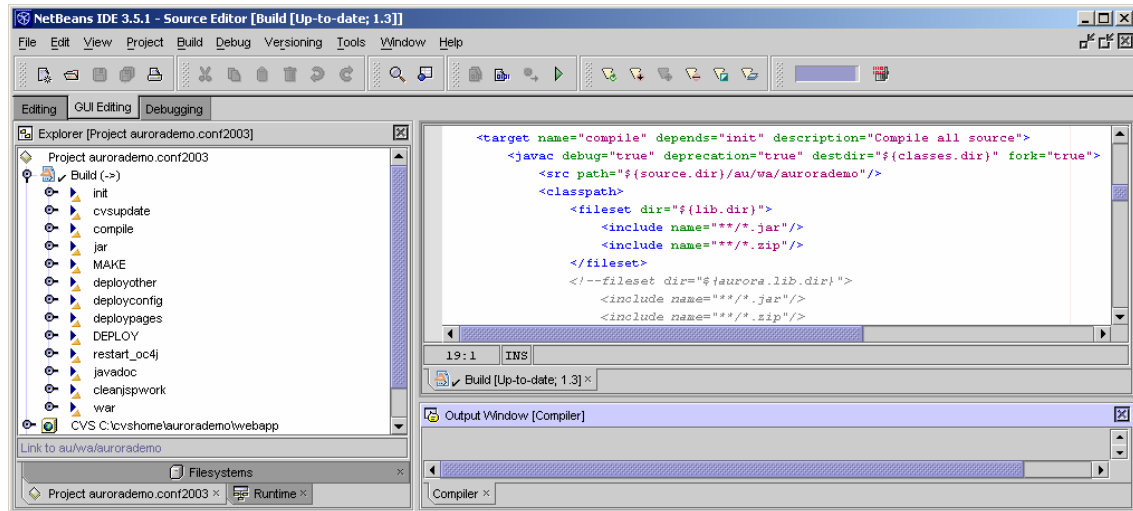
The number in square brackets next to the file name ([1.3] in the above example) indicates the revision number for that particular file. CVS maintains a separate revision number for each file.

Other CVS operations commonly performed in NetBeans are :

- Update. This updates the local Project with all the latest changes from the CVS repository
- Refresh. This refreshes the NetBeans icon and tag to ensure they indicate the correct CVS status.

NetBeans & Ant

Since the Ant Build file is just an XML file, NetBeans recognises its constructs and highlights its syntax. In addition, the developer may execute any Ant target tree by double clicking a target on the NetBean browser window.



Conclusion

As demonstrated in this paper, a full-function and powerful Java IDE may be constructed by just integrating a number of freely available products. Notably, NetBeans is **not** the only free IDE around, other free IDEs are :

- Eclipse (<http://www.eclipse.org>). Originally developed by IBM and donated to the Open Source community.
- JBuilder Foundation (<http://www.borland.com/jbuilder>). Borland offers 3 versions of its JBuilder IDE. The Foundation version, which is the most basic of the 3, may be downloaded for free.

While commercial software may offer features that are not (yet) available in free software, it is interesting to note that there is a large number of Java applications in the commercial world that are developed using free software alone.