CMSC 22610 Winter 2005	Implementation of Computer Languages	Handout 3 January 25

Introduction to CVS

You are expected to keep the source code of your projects in a CVS repository that we will setup for you. If you CS account user ID is `joebob`, then the *root* of you repository will be

```
cvs.cs.uchicago.edu:/stage/cmsc226/students/joebob
```

For each project, we will create a CVS *module*. In the rest of this section, we give a quick introduction to using CVS.

To use CVS, you will first need to set the following environment variables (using **csh** syntax):

```
setenv CVS_SERVER /usr/local/bin/cvs
setenv CVS_RSH ssh
setenv CVSROOT cvs.cs.uchicago.edu:/stage/cmsc226/students/joebob
```

The last of these assumes that you login ID is "`joebob`." Instead of setting the `CVSROOT` variable, it is also possible to use the "`-d`" command line option when issuing CVS commands (see below).

You are now ready to checkout your project. A CVS repository with a module called "`project-2`" will already have been created for you. To *checkout* a copy of this module, run the following command:

```
cvs co project-2
```

or, if you didn't set `CVSROOT`,

```
cvs -d cvs.cs.uchicago.edu:/stage/cmsc237/students/joebob co project-2
```

This command will create a directory called `project-2`. In this directory is another directory called `CVS`, which holds various metadata about this copy of the repository — *you should not change anything in this directory*. You will also find a file named `Makefile`, which we have provided for you. All the files related to your project should live in the `project-2` directory.

Now suppose you create a file called `main.sml` in your `project-2` directory. In order for CVS to keep track of it, it needs to be added to the repository. You do this by the following command:

```
cvs add main.sml
```

You should see a message like:

```
cvs add: scheduling file ‘main.sml’ for addition
cvs add: use ’cvs commit’ to add these files permanently
```

This command records the fact that `main.sml` has been added to the repository, but file will only be added when you commit your changes. To do so, type the following command:

```
cvs commit
```

to add the files permanently to the repository. You will be prompted to enter a log message in an

editor. To specify a particular editor for entering log messages, set the `CVSEDITOR` environment variable. You can also avoid editors altogether by typing your log message on the command line with the `-m` flag:

```
cvs commit -m "added files"
```

After you have entered your message, you will see a message like the following:

```
/tmp/cvsCBrFgq: 9 lines, 337 characters.
Checking in main.sml;
/stage/cmsc237/students/joebob/project-2/main.sml,v  <--  main.sml
initial revision: 1.1
done
```

Changes you make to your files are recorded in the repository every time you do a `cvs commit`. Before you make changes to your files, you can ensure that you have a current version, by running `cvs update`. This fact is not of tremendous significance for individual projects, but matters when more that one person can modify the same files.

Not all the files in your project directory need to be in the repository. For example you should not put your executble files in the repository — these can always be recreated (hopefully!) by compiling the source.

The "`cvs diff`" command is for comparing differences between versions. If no files (or options) are specified, all working files are compared to their last committed versions, otherwise only the specified files are compared. There are also flags to compare other versions, see the man pages or the online manual for details.

The CVS home page is at `www.cvshome.org`. Official documentation is at `www.cvshome.org/docs/manual/`. There is also a nice introduction to CVS at `www.cvshome.org/docs/blandy.html`.