Preliminaries

Written by Lucas Scharenbroich
JPL Machine Learning Systems Group
Copyright 2002, California Institute of Technology.
ALL RIGHTS RESERVED. U.S. Government Sponsorship acknowledged
August 27, 2003

Contents

1 Checking out from CVS 2
2 Starting the Python interpreter 3
1 Checking out from CVS

On the worklab servers, the MLX source repository live in /proj/code. If you are directly using the schema and have no desire to work on the source code, you may skip to Starting up the python interpreter.

Work locally with CVS, one need only define the CVSSROOT environment variable and export it to the shell. The CVSSROOT defines where to tip of the repository lives. Since our code is in /proj/code, the CVS repository is in /proj/CVS. Thus, we need to set our CVSSROOT to /proj/CVS. In bash the command is

```
export CVSSROOT=/proj/CVS/
```

In csh,

```
setenv CVSSROOT /proj/CVS/
```

Now that the CVSSROOT path is set, create a directory in which to check out your copy of the code. A good place might be `/checkout/` which will place the checked out code in a directory called checkout in your home directory. Whatever directory you decide on, lets call it LOCALCODE.

To actually check out a copy of the code, cd to the directory you created (cd $LOCALCODE) and execute the command

```
cvs co code
```

This will check out the entire source code directory which is 60MB. If you are only interested in a portion of the CVS repository, perform something similar to the following

```
cvs co code/python/mlx
```

This will only checkout the code under the code/python/mlx subdirectory. This corresponds to all the python code in MLX.

```
cvs update
cvs log
cvs diff
cvs update
```

Please refer to the CVS resource below for more information on these command. And remember, always do an update before a commit.

A good CVS resource is the cvshome.org website.
2 Starting the Python interpreter

The python interpreter is started by simply invoking the command

```
python
```
rom the command line. For the MLX code base to work, be sure you are using python 2.1 or greater. The python interpreter prints out its version number of startup, so you should see something similar to the following:

```
bash-2.05a$ python
GCC 2.95.4 20011006 (Debian prerelease) on linux2
Type "copyright", "credits" or "license" for more information.
```

The string Python 2.1 indicates we are using version 2.1 of the interpreter. If the version string says Python 2.0 or Python 1.5 you will not be able to use the MLX schema.

Once you are sure that you are using the correct version of python, the next step is to point python to the location of the MLX classes. This is done by setting the PYTHONPATH environment variable. This is similar to the CLASSPATH variable for java interpreters.

If you are currently in the python interpreter (>>> prompt), press ctrl-D to exit back to the shell. If you did not check out a local code of the source code from CVS, set PYTHONPATH to /proj/code/python. In bash:

```
export PYTHONPATH=/proj/code/python/
```

In csh:

```
setenv PYTHONPATH /proj/code/python/
```

If you did check out a local copy, you will need to set your path to $LOCALCODE/code/python/. Again, in bash:

```
export PYTHONPATH=$LOCALCODE/code/python/
```

In csh:

```
setenv PYTHONPATH $LOCALCODE/code/python/
```
Now start up python again (by typing `python` at the command line) and type the following command

```python
>>> from compClust.mix.datasets import Dataset
```

If there is no error, you have set up the python environment correctly. If the interpreter complains that it cannot find `mix.datasets` please review the instruction to ensure that you did not miss anything.

For a complete python reference, go to our local copy.