

NMR NOTES #8

Using the Unix TAR (and BAR) Command

The Unix TAR command is the standard mechanism for archiving data and packing it up into either a named file or magnetic tape cartridge. The BAR command is very similar, but is a SunOS 4.1.x specific command. Files and tapes written by the bar command are not accessible on systems other than SUN systems, and require special handling to extract under Solaris 2.x. This note will detail operation of the TAR command only. See the man page for (many) additional details.

TAR (Tape ARchive) takes a list of files and/or directories, and packs up everything into a single archive. This archive can be directed either to a named file or to a tape drive unit. If directories are named or included, then the command will automatically recurse the directory and archive everything within that directory. The general form of an archive command takes the form:

```
tar cmd [options] file[s] and/or directory[s]
```

The cmd is the specific tar command request, and only one command can be specified. The 5 options are **c**, **r**, **t**, **x**, **u**. The '**c**' command creates a new archive and writes the named files to it. '**t**' lists the Table of contents of an existing archive, so that you can find existing data. '**x**' extracts the named files from an existing archive. If no files are specified, all files in the archive are extracted. The last two commands are not valid on the magnetic tape drives. '**r**' will append data to an existing archive, and '**u**' will update an archive, replacing older files with newer versions. These two commands would not normally be used.

There are several options useful for creating tar files. The **-f** option will let you specify a file name. This can be either a tape drive, specified as **/dev/rmt/0m**, or a named file. Convention is that named files have a **.tar** extension to identify them as tar files. The **-o** option suppresses ownership information from being stored, which may facilitate extracting files on remote systems. More information can be seen by using the **-v** verbose option to give a list of files as they are archived.

A little bit of caution is necessary in specifying the files to be archived. When a file is stored in an archive, the full pathname specified is stored with the data and will be used later to extract the file. This means that if you specify a fully qualified, absolute path name to your data file, it can only be return to that same full path name. For instance, if you execute the command

```
tar -c -f mydata.tar /home2/group/uname/nmrdata/protein/cosy.fid
```

then you will be unable to unpack this data on a remote machine unless it has that same directory structure (**/home/group/uname/nmrdata/protein**). You must also specify that entire pathname if you want to extract just that one file. Consequently, it is much more desirable to specify a relative path, starting at the current directory, or simple file names only. If you gave the comparable commands

```
cd /home/group/uname/nmrdata/protein
```

```
tar -c -f mydata.tar cosy.fid
```

you could then extract the cosy.fid data to your current directory on any system. Also useful is to specify just the directory:

```
cd /home/group/uname  
tar -c -f mydata.tar nmrdata
```

will archive the entire nmrdata subdirectory, so that later if you wanted to extract only the cosy data, you would need to have only an nmrdata subdirectory in your current directory:

```
tar -x -f mydata.tar nmrdata/cosy.fid
```

Using local directory names rather than absolute pathnames retains maximum flexibility for retrieving data later. Avoid starting a file specification with a '/' character.

The last option that may be useful for advanced operations is the -C flag. This directs the system to execute a chdir command to the named file for one file name only. This permits archiving data from multiple subdirectories without having to include full path names. For instance, if you specify

```
tar -c -f mydata.tar -C /home/group/uname/nmrdata  
protein/cosy.fid -C /home/group/user2/nmrdata peptide/cosy.fid
```

you would then have a tar file with two named files: protein/cosy.fid and peptide/cosy.fid.

The tape drive used can be specified as either a command line argument with the -f option or as an environment variable named TAPE. It is generally easier to specify the environment variable rather than type the device each time. The available drives on the nmrsun1 system are:

/dev/rmt/0l 2.5 Gig 8mm drive	/dev/rmt/0ln same, no rewind
/dev/rmt/0m 5 Gig 8mm drive	/dev/rmt/0mn same, no rewind
/dev/rmt/1l 150 Meg drive	/dev/rmt/1ln same, no rewind

Named files are always specified on the command line and by convention should have a **.tar** extension. Also, the tar command and any of the options can be combined, and the - option delimiter is optional. Therefore you can specify

```
tar cf mydata.tar mydata.fid
```

is the same as

```
tar -c -f mydata.tar mydata.fid
```

After creating a tar file, you may also want to use the compress program to compress the file and reduce its size. The command compress mydata.tar will replace the file mydata.tar with the smaller mydata.tar.Z. Finally, the Gnu utility gzip is another alternative that will compress it even more. gzip mydata.tar will create the file mydata.tar.gz, which will normally be smaller than the mydata.tar.Z file. Both the .Z and the .gz extensions are automatically supplied. The gunzip command is used to expand .gz files, and uncompress will expand .Z files. Beware that gzip/gunzip may not be available at all site, but the compress/uncompress are standard UNIX utilities.