CODE MAINTENANCE AREAS ON VLBACC

B. Clark & R. Heald

September 1989

It is proposed that we maintain the VLBA station control software using SCCS. This has a couple of minor disadvantages for maintaining code and test areas (e.g. the audit trail will not be part of the source code), and relating to the code repository in Charlottesville, but they seem to be tolerable. Things get a little complicated when you think about it in detail. Here are some suggestions about how we do things.

/home/vw/ would contain only Wind River supplied code.

I suggest the following structure for the vlbacc equivalents of the MicroVax VLBSOFT area.

vlbsoft/mv121	The vxWorks OS code customized for the Motorola MVME121 Microprocessor. This includes code for the MVME121 ROM. This single area contains source, object, Motorola S-records, and scripts.
vlbsoft/code	The code currently operating at most of the VLBA stations; the code residing on their local disk. Includes source code, object code, object libraries (for linking Sun programs), Sun executables, and scripts for both VxWorks and Sun.
vlbsoft/test	The code normally used at PieTown (booted by wire if it does not take too long, with the advantage of ease of bringing up the vlbsoft/code version in case of trouble)
vlbsoft/h	Header files of the same provenance as /test
vlbsoft/codeNew	Code submitted for update, but not used for normal observing; should be used for test observing, especially at Pie Town, as much as possible to complete debugging.
vlbsoft/hNew	Header files of the same provenance as /codeNew
vlbsoft/data	Station data (e.g. pointing parameters), device data (e.g. transport specific parameters), and sample/archetype observe files. Files managed by SCCS.
vlbsoft/doc	Home for synopsis.txt and other multimodule documentation material
vlbsoft/SCCS	SCCS work area, best ignored by human beings.
vlbsoft/doc/work	exdoc work area, ditto.

The directories code, test, and codeNew would have subdirectories code/ccc, test/ccc, and codeNew/ccc which would contain copies of the source code modules maintained on ccc. The object modules would be in the main directory, to minimize the number of libraries searched. For the same reason there is not a ccc subdirectory for .h files; just because a .h file exists in those directories, do not assume you can get it for editing with an SCCS edit command.

Programers should have a logical link to vlbsoft/SCCS in their working directory so that they may directly submit code merely by saying "sccs create <filename>" (or "delta <filename>", etc.).

It is strongly urged that sccs identifiers be included in all files. Including the file vlbsoft/doc/byline.txt somewhere in the first few lines of the file is sufficient.

In order that the above can be maintained more or less automatically, it is suggested we adhere strictly to a set of filename suffixes:

none	= linked or partially linked executables in a.out format
.c	= C source
.csh	= Sun C-shell scripts
.dat	= data
.h	= C include (vendor supplied)
.H	= VLBA C include (name is in all caps)
.il	= assembler function expansions

.o = assembled/compiled module in relocatable a.out format
.s = assembler source
.sh = VxWorks shell scripts
.txt = documentation

The operators would run Sun programs from vlbsoft/test except, in case of trouble, from vlbsoft/code, or for test observing, vlbsoft/codeNew.

The lab computer would have the code from vlbsoft/test on its local disk, or could be booted by wire from vlbsoft/program.

When a station computer is turned on, when the front panel reset button is pressed, or when the "reboot" command is executed by the vxWorks shell the following sequence would occur:
1) The vxWorks ROM monitor is initialized and executed. This program loads vxWorks and begins its execution. By default, it will load from the hard disk; through manual intervention it can load from the floppy in case hard disk failure, or possibly by wire in case of controller failure. We should keep a system floppy at each station for emergency use.
2) vxWorks' first action is to execute the script file "startup.cmd" in the same directory from which vxWorks was booted. This would contain two pieces of information: the files to be loaded (in order), and where the loading process is to be stopped. The reasonable stopping points are, a) immediately, b) after loading NRAO supplied infrastructure routines (e.g. math routines, mcbio()), c) after loading the screens package support routines, d) after loading the real-time observing suite, e) after loading the screens themselves, f) after starting the observing tasks, g) at completion, after starting the top level screen and suspending the shell.

The proposed operation for maintaining VLBA software: programers use SCCS in its approved manner, primarily through the SCCS edit and SCCS delta commands. SCCS fix would be discouraged (I suggest limiting it to same-day, certainly to same week fixes). Multiple file SCCS delta would be discouraged but not forbidden, on the grounds that it is often useful to have slight differences in the audit trail from file to file that doesn't occur to one unless faced on an individual basis. (Having the audit trail generated in real time rather than by an editor is one of my minor annoyances with SCCS).

Then, probably by an automated "midnight job", freshly submitted code would be SCCS get'ed into vlbsoft/hNew and vlbsoft/codeNew, and the appropriate make executed. .dat files would be appropriately squeegied and copied to the appropriate stations.

Once a month, vlbsoft copies source code from vlbsoft/codeNew into vlbsoft/test and makes the entire system, including the documentation extraction. He then deletes everything from vlbsoft/codeNew.

About a week later, after problem-causing modules have been fixed or deleted, the executables would be copied from vlbsoft/test onto floppies and mailed to the stations (or it may be acceptable to send them by wire) to be copied to the station local disks. At the same time, the contents of vlbsoft/test would be copied to vlbsoft/code and then deleted from vlbsoft/test.

I use synopsis.txt heavily, and progindex.txt occasionally. I have never found any use for the .doc files. Unless I hear otherwise, I propose to modify "exdoc" to remove them from the system.