22. SOFTWARE MAINTENANCE PROCEDURES

This section outlines maintenance procedures for selected pieces of software that are in fact still maintained at all. The primary intent is to document the location of the sources and to identify the current maintainers.

22.1. Midas

Maxc2 only. The current sources for Midas are kept on the "Maxc2 Midas" Alto disk (maintained by Fiala). Reasonably current sources are also kept on [IVY]<MAXC] or [MAXC1]<MAXC> directories: MSOURCES.DM along with MIDAS.RUN, LOADER.MB, and DEBUG.MB. The various Midas command files are kept on DGSOURCES.DM along with the sources for the microdiagnostics.

Before initializing Midas on a new disk, be sure that you have the following files ready:

Midas.Programs	(see below)
Gacha10.al	The font Midas uses
Midas.Midas	The initialization command file

Midas.Programs contains a list of file names separated by blanks, commas, or carriage-returns. The names must be typed in UPPER-CASE. This list serves two purposes. First, file pointers are built for all of the names to speed up OpenFile. Next, the list of names for the "Run-Program" command menu is built. If the file name contains no extension, then hint FP's will be built for both name.MB and name.MIDAS and name will be put in the "Run-Program" menu. (However, the hint FP's are not built unless the file exists, and the file name will not be put in the "Run-Program" menu unless name.MIDAS exists.) If the file name contains an extension, then it will be put in the quick OpenFile table, but won't appear in the "Run-Program" menu.

Midas creates and uses the following files:

Midas.State	~40 pages	Built for quick init by Midas/I
Midas.Resume	~40 pages	Built before calling AltIO for resume
Midas.FixUps	2 pages	Built when external fixups occur in .MB files being loaded (Current microcode never uses this.)
Midas.Errors	2 pages	Written when "Compare" fails
Midas.SymTab	100 pages	Written when symbol table overflows core buffers

Altogether this is about 200 disk pages. It is desirable to do this with a pretty clean disk, so that the files wind up contiguous on the disk and near to each other.

22.2. NVIO

Maxc1 only. NVIO is maintained by Taft using a Nova disk labelled "NVIO". Backup copies of the source files are kept in the <NVIO> directory on Maxc1.

The NVIO source files all have the extension .NS, while command files for assembling and loading these sources have extensions .AS and .CF. Instructions for building a new version of NVIO are included as comments at the beginning of the source file C.NS.

22.3. AltIO

Maxc2 only. AltIO is maintained by Taft using an Alto disk labelled "AltIO/Maxc2". Backup copies of the source and command files are kept in a single dump-format file <MAXC>ALTIO.DM on Ivy.

AltIO may be compiled and loaded by executing the command files CompileAltIO.cm and LoadAltIO.cm respectively. AltIO makes use of a number of other Alto software packages available from the <Alto> directory. These packages are listed in LoadAltIO.cm.

22.4. TM, MemBash, SMIDiag, Alto Microcode

Maxc2 only. These pieces of software are kept as dump-format files in the <Maxc> directory on Ivy. The files are TM.dm, MemBash.dm, SMIDiag.dm, and MaxcAltoCode23.dm. They are compiled and loaded by means of command files whose names follow the conventions used for AltIO.

22.5. Tenex and Diagnostic Microcode

Sources for Maxc1 and Maxc2 Tenex microcode are maintained on the "Maxc1/2 Tenex Microcode" Alto disk by Fiala. Sources for Maxc1 and Maxc2 microdiagnostics are maintained on the "Maxc1/2 Microdiagnostics" Alto disk by Fiala.

These are backed up by Alto dump files stored on the IVY <MAXC> directory. Common sources are used for both Maxc1 and Maxc2 with conditional assemblies used for parts of the microcode that are different on the two systems.

Cross reference listings of Tenex microcode sources are obtained by FTPing the sources to the Maxc1 ERF directory and doing RUNFIL CRREF.RUNFIL, which uses MCROSS to produce the cross reference listing. The same procedure is used for microdiagnostic sources except that the DCRREF.RUNFIL command file is used.

22.6. Tenex

The Tenex systems for both Maxc1 and Maxc2 are generated from a common set of sources maintained on Maxc1 in the <134> directory (for Tenex version 1.34). Differences between the two systems are dealt with by conditional assembly.

The RUNFIL scripts LOADMAXC1.RUN and LOADMAXC2.RUN may be used to assemble and load new versions of Tenex. Most of the source files are separately assembled for each system, yielding .REL files with extensions .1RL and .2RL. However, some of the assemblies do not depend on any system parameters, so Maxc1 and Maxc2 share common .REL files for these. When changes are made to any system parameters in PARAMS.MAC or PROLOG.MAC, all the .1RL and .2RL files should be deleted to force reassembly of them.

The result of assembling and loading a new Tenex is a file called MAXC1.SAV or MAXC2.SAV, plus some auxiliary files whose extensions are .MAXC1 or .MAXC2. The bug string file and loader map are also printed on Ears; these should be placed next to the log book in the Maxc room.

After loading a new Tenex for Maxc1, all that is necessary is to run Micro-Exec and install MAXC1.SAV on an unused save area (e.g., 19) for stand-alone checkout. The "Read.Tenex.From.-File" and "Write.Tenex.To.Area" commands should be used for this purpose.

After constructing a new Tenex for Maxc2, one should transfer the various required files to Maxc2 by means of PUPFTP. These files are <134>MAXC1.SAV, <SYSTEM>BUGTABLE.MAXC2 and <SYSTEM>MONSYMS.MAXC2. These files should be transferred to the same directories and preserving the version numbers. They may then be deleted on Maxc1. Then the file MAXC2.SAV should be installed on a Maxc2 save area as explained above.