MEMMAN

Voyager CCS Memory Management Software

Software Maintenance Document

Revised May 3, 1990

Jet Propulsion Laboratory

Gregory F. Welch

Table of Contents

1. Statement of Purpose

The purpose of this document is to attempt to give the reader specific steps to follow in order to update the MEMMAN program for reasons such as failure correction or software enhancement. Among other topics, the user will learn how to access the source code files, edit them, re-compile the program and redeliver it.

It is assumed that the reader has an understanding of the functions of MEMMAN from a SEQTRAN/COMSIM analyst's perspective, and that they are familiar with the operation of the program. This document offers a basic overview of the software operation, but in no way attempts to provide instructions for setting up and using the software. See MEMMAN Functional User's Guide[1] for a more detailed description of program operation.

It is also assumed that the reader has sufficient knowledge of Pascal in general, and the Turbo Pascal (TP) environment in particular. This document will make no attempt to explain Pascal or the Turbo Pascal system. For such background, the user should refer to the Turbo Pascal Owner's Handbook[2].

2. Introduction

Although MEMMAN was originally written using Borland International's Turbo Pascal (TP) version 4.0, later updates were implemented using version 5.0 in order to take advantage of features available only in the new version (mainly the debugging features). There have been newer releases of TP since 5.0, but MEMMAN has never been updated/maintained using any of these newer versions. To do so would require the user to study Borland's documentation of differences between the two versions, and possibly modify the source code to be compatible with the newer version.

All of the maintenance instructions in this document are based on the personal computer configuration discussed in 5. Maintenance Environment. If the reader wishes to change disk drive names (locations) or directory conventions, it is up to them to interpret the instructions in this document which refer to specific disk drives or directories.

In order to begin maintenance of the MEMMAN program, the reader should first read sections 3, 4 and 5 in order to configure their personal computer correctly. The reader should then study section 6 in order to understand how the source code is stored and accessed. Finally, by referring to sections 7 through 12 the reader should be able to successfully edit, compile and deliver the MEMMAN software.

3. Hardware Requirements

In order to maintain MEMMAN, there is a minimum set of hardware which must be available. In addition to describing this minimum hardware set, the following table describes the hardware actually used during MEMMAN development. It is the "Developed On" system which will be referred to in this document, and any variances should be interpreted by the reader.

MINIMUM	DEVELOPED ON
IBM PC-AT	IBM PC-AT Compatible (Everex System 1800)
640 KByte RAM	640 KByte RAM (+1.5 MByte Expanded)
20 MByte fixed disk drive	40 MByte fixed disk drive (20 as C:, 20 as D:)
CGA graphics adapter (color)	Color VGA (Paradise Professional)
CGA monitor (color)	Color VGA (NEC Multisync II)
Internal clock/calendar	Internal clock/calendar
5 1/4" 1.2 MByte floppy disk drive	5 1/4" 1.2 MByte floppy disk drive (A:) 5 1/4" 360 KByte floppy disk drive (B:)
Medium-high speed dot matrix printer	Epson LQ1500
Math Coprocessor	80287 Math Coprocessor

4. Software Requirements

The following software is required to maintain MEMMAN per the instructions of this document. It is possible to make substitutions or in some cases omissions, but that will not be discussed in this document. For more information on the PKARC and PKXARC utilities, see Appendix B: More About PKARC & PKXARC, on page 24.

Located in C:\ and C:\DOS

MSDOS (Microsoft's Disk Operating System, Version 3.3)

COMMAND.COM	DOS program
(Misc. files)	Associated MSDOS programs & data files

Located in D:\TP

Turbo Pascal (Borland International's Pascal language, Version 5.0)

TURBO.EXE	Turbo Pascal 5.0 program
(Misc. files)	Associated Turbo Pascal (TP) program & data files
LISTER.EXE	Source code listing program LISTER.PAS compiled with TP (LISTER.PAS is included with the TP system)

Located in C:\ARC

PKARC (FAST! Archive Create/Update Utility, Version 3.5, 04-27-87)

PKARC.COM	Archive create/update utility program
PKARC.DOC	Archive create/update utility documentation

PKXARC (FAST! Archive Extract Utility, Version 3.5, 04-27-87)

PKXARC.COM	Archive extraction utility program
PKXARC.DOC	Archive extraction utility documentation

5. Maintenance Environment

In discussing the MEMMAN maintenance, it will be assumed that the user has established an operating system environment which matches the directory and path structures on the MEMMAN development system, discussed below. Variations from this environment are possible if the reader factors these variations into the rest of this document, with particular attention being given to the batch files seen beginning on page 17 (Appendix A: Batch Files).

5.1 Disk Partitions

As noted on page 3 (3. Hardware Requirements) the MEMMAN development system had a single 40 MByte fixed disk which was partitioned into two separate logical drives called C: and D:. The C: drive hosts the individual sequence files and the delivered MEMMAN system with all of the necessary files, while the D: drive hosts all of the necessary maintenance software and source code archives.

5.2 Directory Structure

As can be seen in Figure 1, each sequence has a subdirectory under the subdirectory SEQ. Each of the individual sequence directories contains any needed files for that particular sequence, such as any input & output SRFs, configuration files, runstream files, memory map files, and other miscellaneous files.

The MEMMAN system is located in its own subdirectory under the SEQ-TOOL (sequence tool) subdirectory. This MEMMAN subdirectory contains the delivered MEMMAN program as well as all of the needed support files such as the catalog file, the system parameter file, etc....

Notice also that there are separate subdirectories for all of the batch files (BATCH), and all of the DOS related files & programs (DOS). Such a directory structure allows the use of batch files to begin several processes, without an extensive path variable setting (see 5.3 DOS Path below).

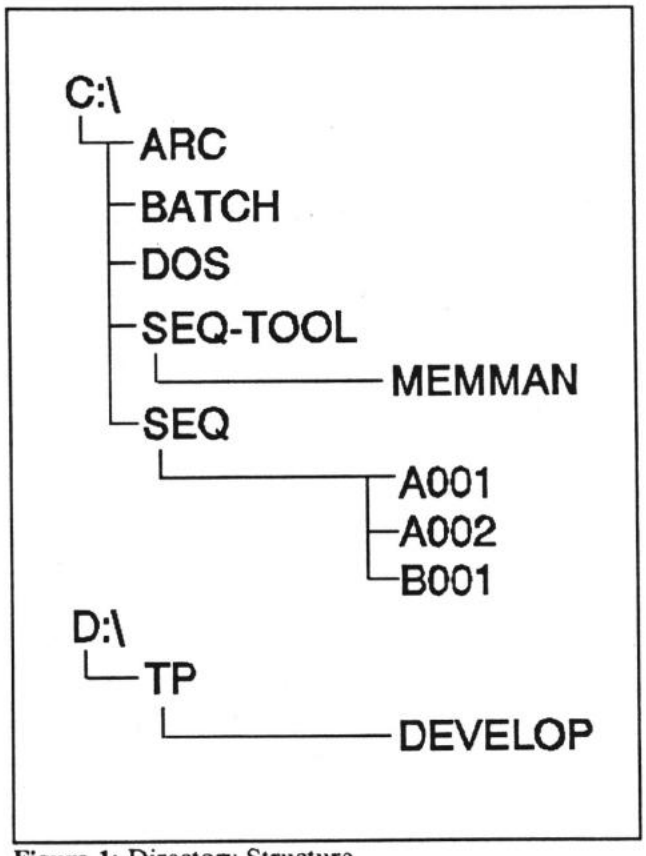

Figure 1: Directory Structure

5.3 DOS Path

The DOS shell variable *Path* determines the path the operating system will follow when looking for executable programs. In order to use the batch files and procedures outlined in this document, the system path must include the DOS and the BATCH directories.

For example, the following DOS path command should appear in the automatically executed batch file AUTOEXEC.BAT (see the IBM DOS Reference Manual[3] for further information about AUTOEXEC.BAT):

```
PATH=C:\BATCH;C:\DOS;
```

Such a path setting will cause the operating system to begin searching for executable programs in the two directories BATCH and DOS. If a path setting already exists in the AUTOEXEC.BAT file, then the directories BATCH and DOS should be included in that path setting (e.g. append the text `C:\BATCH;C:DOS;` to the end of the existing path).

6. Source Code Archives

All of the source code for MemMan is stored in archive files which are created by the utility program PKARC which collects and compresses many files into a single archive file (see also B. More About PKARC & PKXARC). There are two archive files in particular which pertain to MEMMAN; MAN_PAS.ARC and MISC_PAS.ARC, and both should be located in the D:\TP\DEVELOP directory (shown in Figure 1). The contents of these two archive files are described in 6.1 Archive Contents.

When a file is to be modified, the files must be unpacked from the archive (using the program PKXARC) and then edited. When the edits are complete, the modified routine should be packed back into the archive (using PKARC) and the archive should be backed-up to a floppy diskette. The following sections will describe the archives and how to manipulate them for MEMMAN maintenance. For more information on editing the source code files, see 8. Editing Source Code, and for more information on backing up the archives see 6.6 Backing-Up an Archive.

6.1 Archive Contents

As discussed previously, there are two archive files in particular which pertain to MEMMAN; MAN_PAS.ARC and MISC_PAS.ARC. As is indicated by the archive names, MAN_PAS.ARC contains all of the MEMMAN source code (TPU source code) and data files (copies of actual data files), and MISC_PAS.ARC contains source code for any miscellaneous TPUs which are not used exclusively by MEMMAN.

The following tables list the contents of the MAN_PAS.ARC and MISC_PAS.ARC archive files, along with their approximate sizes (as of the printing of this document).

Contents of MAN_PAS.ARC (193973 Bytes):

MAN_ALST.PAS	2525 Bytes	MAN_AUTO.PAS	30676 Bytes
MAN_BOXS.PAS	1984 Bytes	MAN_CAT.PAS	16497 Bytes
MAN_CURS.PAS	11458 Bytes	MAN_CYCL.PAS	4492 Bytes
MAN_ERRS.PAS	20656 Bytes	MAN_FORC.PAS	14964 Bytes
MAN_GCNT.PAS	26873 Bytes	MAN_GEN.PAS	55069 Bytes
MAN_HELP.PAS	3166 Bytes	MAN_HIST.PAS	17111 Bytes
MAN_INIT.PAS	8226 Bytes	MAN_LOAD.PAS	57238 Bytes
MAN_LOG.PAS	4322 Bytes	MAN_MAIN.PAS	9053 Bytes
MAN_MANL.PAS	30177 Bytes	MAN_MISC.PAS	53936 Bytes
MAN_PCMD.PAS	45720 Bytes	MAN_PLAT.PAS	2363 Bytes
MAN_SCR1.PAS	35223 Bytes	MAN_SMDF.PAS	16544 Bytes
MAN_SPEC.PAS	1321 Bytes	MAN_SRCH.PAS	4653 Bytes
MAN_SYS.PAS	13522 Bytes	MAN_TST.PAS	792 Bytes
MAN_VARS.PAS	43078 Bytes		
MEMMAN.CAT	2829 Bytes	MEMMAN.LOG	370 Bytes
MEMMAN.SMD	6714 Bytes	MEMMAN.SYS	203 Bytes

Notice that the total size of the individual files is 541,755 Bytes, but the size of the archive file is only 193,973 Bytes. This is one reason the archive utility is used, along with the fact that it allows the compartmentalization of files into similar groups.

Contents of MISC_PAS.ARC (52826 Bytes):

AUXINOUT.PAS*	8379	Bytes	AUX_INTR.PAS*	15250	Bytes
COLORS.PAS*	303	Bytes	COMWATCH.PAS*	1023	Bytes
EM_MGR.PAS*	6758	Bytes	GW_INPUT.PAS	58115	Bytes
GW_LIB.PAS	17742	Bytes	MOUSE.PAS	20081	Bytes
WINDOWS.PAS	14468	Bytes			

* Not actually used by MEMMAN

Notice again that the total size of the individual files is 142,119 Bytes, but the size of the archive file is only 52,286 Bytes. Also here notice that several of the source code files contained in the MISC_PAS.ARC archive are not used by MEMMAN. This is because the archive MISC_PAS.ARC contains miscellaneous source code files for *any* TP program, not necessarily files just for MEMMAN.

6.2 Listing Files in an Archive

In order to list the files in an archive, the PKARC program must be called with a special parameter (besides the archive name). This special running of PKARC has been incorporated into a batch file called LIST.BAT (see A.4 List). Therefore, in order to list the contents of an archive, the user need only to change to the directory containing the archive, and at the DOS prompt type:

`LIST` *archivename[.ARC]*

This will invoke the batch file LIST.BAT with the parameter *archivename* (notice that the extension ARC is optional). This batch file will then call PKARC with the correct parameters to list the archive file. Notice that if no parameter (*archivename*) is supplied when LIST.BAT is run, the batch file will notify the user of the correct syntax.

6.3 Unpacking a File From an Archive

In order to extract (unpack) a file (or files) from an archive, the PKXARC program must be called with the correct parameters. This special running of PKXARC has been incorporated into a batch file called UNPACK.BAT (see A.3 Unpack). Therefore, in order to unpack a file (files) from an archive, the user need only to change to the directory containing the archive, and at the DOS prompt type:

`UNPACK` *archivename[.ARC] filename(s)*

This will invoke the batch file UNPACK.BAT with the parameter *archivename* (notice that the extension ARC is optional) and the desired *filename*(s) to unpack. This batch file will then call PKXARC with the correct parameters to unpack the file(s) from the archive file. Up to eight file names can be supplied with the batch file call, and all eight will be unpacked. In addition, the filename parameter can also be a file mask, in order to extract all files which meet a certain criteria, or it can be omitted completely and the entire contents of the archive will be extracted.

Notice again that if no *archivename* is supplied when UNPACK.BAT is run, the batch file will notify the user of the correct syntax.

6.4 Packing a File in an Archive

In order to pack (update) a file (or files) in an archive, the PKARC program must be called with the correct parameters. This special running of PKARC has been incorporated into a batch file called PACK.BAT (see A.2 Pack). Therefore, in order to pack (update) a file (files) in an archive, the user need only to change to the directory containing the archive, and at the DOS prompt type:

PACK *archivename[.ARC] filename(s)*

This will invoke the batch file PACK.BAT with the parameter *archivename* (notice that the extension ARC is optional) and the desired *filename*(s) to pack. This batch file will then call PKARC with the correct parameters, which will add the file(s) to the archive if they aren't already there or update them if they are. Up to eight file names can be supplied with the batch file call, and all eight will be packed (or updated). In addition, the filename parameter can also be a file mask, in order to pack all files which meet a certain criteria.

Notice again that if no *archivename* is supplied when UNPACK.BAT is run, the batch file will notify the user of the correct syntax.

6.5 Removing a File From an Archive

On a rare occasion it may be desirable to remove a file from within an archive. This may be the case if a file was accidently added to the archive, or if it is determined that one is no longer needed (for whatever reason). In order to delete a file (or files) from an archive, the PKARC program must be called with the correct parameters. This special running of PKARC has been incorporated into a batch file called DELETE.BAT (see A.5 Delete). Therefore, in order to delete a file (files) from an archive, the user need only to change to the directory containing the archive, and at the DOS prompt type:

DELETE *archivename[.ARC] filename(s)*

This will invoke the batch file DELETE.BAT with the parameter *archivename* (notice that the extension ARC is optional) and the desired *filename*(s) to delete. This batch file will then call PKARC with the correct parameters, which will remove the file(s) from the archive if they are there. Up to eight file names can be supplied with the batch file call, and all eight will be deleted. In addition, the filename parameter can also be a file mask, in order to delete all files which meet a certain criteria.

6.6 Backing-Up an Archive

After MEMMAN has been updated (modified) and the software is ready to be delivered, any modified source code files should be packed into the archive files, and the archive files should be backed-up on one of the MAN_PAS (or MISC_PAS) archive file backup disks (a floppy disk). To facilitate this process, the user can use the batch file BACK_PAS.BAT in order to create a backup of the archive file on a floppy disk.

In order to backup either the MAN_PAS.ARC or the MISC_PAS.ARC archive files, follow these steps:

1) Place the 5 1/4" 1.2 MByte archive backup disk (either MAN_PAS or MISC_PAS) in drive A: (Make sure the disk has enough space available to store the archive).
2) At the DOS prompt...

a) Type `D:`
b) Type `CD \TP\DEVELOP`
c) Type `BACK_PAS` *archivename date*

archivename is the archive file name (i.e. MAN_PAS.ARC or MISC_PAS.ARC) and *date* is today's date in the form MM-DD-YY.

3) Watch the screen as the archive file is copied and the CHKDSK program (DOS program) is run automatically (make sure that the floppy disk is not full).

Following these steps will provide the user with a new directory (named today's date) on the backup floppy disk, and a copy of the latest archive file in that directory. Following these steps at the end of every day [when updates to the source code are made] will ensure that a copy of every version of the code is kept for later reference or restoration (if needed).

7. Maintenance Steps

The following general steps would normally be followed to complete an update to the MEMMAN program:

1) Change to the D: drive.
    a) From the DOS prompt, type `D:`
2) Change to the directory containing the source code.
    a) Type `CD \TP\DEVELOP`
3) Unpack all of the necessary source code & data files from the MAN_PAS.ARC and MISC_PAS.ARC archives.
    a) Type `UNPACK MAN_PAS`
    b) Type `UNPACK MISC_PAS`
4) Run the Turbo Pascal program [batch file] (see A.1 Turbo).
    a) Type `TURBO`
5) Complete any edits & debugging (see sections 8 through 11).
6) Complete the program delivery. See 12. Delivery of Executable.

8. Editing Source Code

The following steps would normally be followed to edit any of the MEMMAN related Pascal source code files (either MEMMAN files from MAN_PAS.ARC or miscellaneous files from MISC_PAS.ARC):

1) If not already done, run the Turbo Pascal program [batch file] (see A.1 Turbo).
   a) Type TURBO
2) Change to the directory where the source code is located (D:\TP\DEVELOP).
   a) Press <Alt><F> (both keys together) to pull down the *File* menu
   b) Press the <C> key to select *Change Directory*
   c) Press the <End> key to move the cursor to the end of the line
   d) Type \DEVELOP and press <Enter>
3) Load the desired file into the TP editor
   a)* Press <Alt><F> (both keys together) to pull down the *File* menu
   b)* Press the <L> key to select *Load*
   c) Press <Enter> to select the default file list mask of *.PAS
   d) Using the arrow keys, highlight the desired file and press <Enter>

   * Steps a) and b) can also be accomplished by pressing the <F3> key.
4) Perform the necessary edits, using the editor commands described in the Turbo Pascal Owner's Handbook and the <F1> help screen (similar to Wordstar editor commands).
5) Save the edited source code file
   a)* Press <Alt><F> (both keys together) to pull down the *File* menu
   b)* Press the <S> key to select *Save*

   * Steps a) and b) can also be accomplished by pressing the <F2> key.

The above steps should be repeated until all of the desired files have been changed. At that point, usually the user would then compile the program in order to check for errors (see 11. Compiling).

9. The GREP Utility

The Turbo Pascal software package comes with a very handy utility program called GREP. The GREP program provides the user with the capability to search multiple files for any occurrences of a particular string of characters. Although the utility is described in detail in the Turbo Pascal Owner's Handbook, a simple (typical) example is shown below.

For instance, what if the user wanted to search all of the MEMMAN related Pascal source code files (located in the D:\TP\DEVELOP directory) for any calls to a procedure named ProcessCommand? To use GREP to accomplish this, the user would follow these steps:

1) Change to the directory containing the Pascal source code files.
   a) From a DOS prompt type `D:` to change to the D: drive (if needed)
   b) Type `CD \TP\DEVELOP` to change to the desired directory
2) Run the GREP utility program.
   a) Type `\TP\GREP -i ProcessCommand *.PAS`

      `\TP\GREP` runs the GREP utility, located in the \TP directory
      `-i` instructs the utility to ignore upper or lower case differences
      `ProcessCommand` specifies the string to search for
      `*.PAS` specifies the search file mask

3) Watch the screen closely as each occurrence of the string will be identified while GREP searches all of the target files (specified by the search file mask *.PAS).

If the utility finds any occurrences of the string in a file, it will [on the screen] list the file name, followed by each line from the file which contains the target string. Along with the `-i` option, there are many other options which control what the utility looks for and how it displays the results. Consult the Turbo Pascal Owner's Handbook for further information.

10. Printing Source Code

In order to print a Pascal source code file, virtually any method of printing can be used (e.g. the DOS Print command, etc...). However, the TP software package comes with the source code for a TP program called LISTER (source code LISTER.PAS). This program will not only neatly handle page breaks while printing, but it will also "read in" any source code files specified in an embedded $I (include file) directive.

In order to use the LISTER program however, the user must compile the program (preferably to disk). In order to compile the LISTER program (to disk), follow these steps (skip these steps if the executable program LISTER.EXE is already available):

1) Change to the directory containing the Pascal system files.
   a) From a DOS prompt type `D:` to change to the D: drive (if needed)
   b) Type `CD \TP` to change to the desired directory
2) Copy the LISTER.PAS source code to the D:\TP directory.
   a) Insert the appropriate TP disk in drive A:
   b) Type `COPY A:LISTER.PAS D:\TP\LISTER.PAS`
3) Run the Turbo Pascal program [batch file] (see A.1 Turbo).
   a) Type `TURBO`
4) Change the *Primary File* option to specify the LISTER program, and assure that the compile *Destination* option is to disk.
   a) Press <Alt><C> (both keys together) to pull down the *Compile* menu
   b) Press the <P> key to select *Primary File*
   c) Type `LISTER.PAS` and press <Enter>
   d) If the *Destination* option is set to Memory, then press the <D> key to toggle it to Disk
   e) Press the <Esc> key to clear away the *Compile* menu
5) Compile the LISTER program (using the Make option).
   a)* Press <Alt><C> (both keys together) to pull down the *Compile* menu
   b)* Press the <M> key to select *Make* and press <Enter>
   c) Wait for the compilation to complete, and then press any key (as instructed)

   * Steps a) and b) can also be accomplished by pressing the <F9> key.
6) Exit the Turbo Pascal program (return to DOS prompt).
   a)* Press <Alt><F> (both keys together) to pull down the *File* menu
   b)* Press the <Q> key to select *Quit*

   * Steps a) and b) can also be accomplished by pressing the <Alt> and <X> keys simultaneously.

Once at a DOS prompt, the following steps would normally be followed to print any Pascal source code files:

1) Change to the directory containing the Pascal source code files.
   a) From a DOS prompt type `D:` to change to the D: drive (if needed)
   b) Type `CD \TP\DEVELOP` to change to the desired directory
2) Run the LISTER program.
   a) Type `\TP\LISTER`
   b) Type the name (including the extension .PAS) of the source code file to be printed and press <Enter> (e.g. type `MAN_MAIN.PAS` and press <Enter>)

11. Compiling (Building)

In general, the TP system provides three separate methods of compiling (from within the TP environment); *compile* a single source code file, *make* a program, or *build* a program. Using the regular *compile* option will only compile the source code which is currently loaded in the editor. If that source code uses another TP unit which is not up to date (has not been re-compiled after a modification) then an error will occur and the compilation will terminate. In other words, every TPU which is used by the main program (the one in the editor) must be independently brought up to date or the compile will not work. At the other extreme is the *build* option. This will compile (re-compile) every TPU used by the primary file, regardless of whether it needs to be re-compiled or not.

In general, the *make* compile option should be used to compile the MEMMAN program. This option, which is in a sense midway between the *compile* and *build* options, will compile the program named as the *Primary File* in the *Compile* pull-down menu, and it will compile (re-compile) any needed TPUs which are not up to date (modified since their last compile). This feature frees the user from the task of managing the individual units and their compiled states, and provides the quickest compilation (on the average).

In order to compile the MEMMAN program, follow these steps:

1) If not already done, run the Turbo Pascal program [batch file] (see A.1 Turbo).
   a) Type TURBO
2) Change to the directory where the source code is located (D:\TP\DEVELOP).
   a) Press <Alt><F> (both keys together) to pull down the *File* menu
   b) Press the <C> key to select *Change Directory*
   c) Press the <End> key to move the cursor to the end of the line
   d) Type \DEVELOP and press <Enter>
3) Assure that all of the TP settings match those given in Appendix C: Turbo Pascal Settings.
4) If any of the settings were modified, save the settings.
   a) Press <Alt><O> (both keys together) to pull down the *Options* menu
   b) Press the <S> key to select *Save options* and <Enter> to accept the default options file name (Turbo.TP)
5) Start the compilation.
   a)* Press <Alt><C> (both keys together) to pull down the *Compile* menu
   b)* Press the <M> key to begin the *make* process

   * Steps a) and b) can also be accomplished by pressing the <F9> key.
6) If it was modified, save the file in the TP editor
   a)* Press <Alt><F> (both keys together) to pull down the *File* menu
   b)* Press the <S> key to select *Save*

   * Steps a) and b) can also be accomplished by pressing the <F2> key.

12. Delivery of Executable

At the present time, there is no paper work involved with the delivery of the MEMMAN program. In fact, the delivery of the MEMMAN executable is fairly straight forward. To deliver the MEMMAN executable, follow these steps:

1) If not already done, update the MEMMAN version.
   a) Edit the file MAN_VARS.PAS (see 8. Editing Source Code)
   b) Press <Ctrl><Q> and then <F> to find the *Version* constant
   c) Type `VERSION` as the text to search for
   d) Type `GUN` for the search options
   e) Change the *Version* value (increase) and the date in the comment
   f) Save the file (MAN_VARS.PAS)
   g) Re-compile the MEMMAN program (see 11. Compiling)
   h) Exit the TP program
2) Re-pack any changed source code files into the MAN_PAS.ARC and MISC_PAS.ARC archives and then delete all of the source code & data files (see A.6 Clean).
   a) Type `CLEAN MAN`
   b) Type `CLEAN MISC`
3) If the MAN_PAS.ARC archive was updated (as a result of step 2 above) then back-up the MAN_PAS.ARC source code archive (see A.7 Back_Pas).
   a) Insert the MAN_PAS archive floppy disk in drive A:
   b) Type `BACK_PAS MAN 05-01-90` (if today's date were May 1, 1990).
4) If the MISC_PAS.ARC archive was updated (as a result of step 2 above) then back-up the MISC_PAS.ARC source code archive (see A.7 Back_Pas).
   a) Insert the MISC_PAS archive floppy disk in drive A:
   b) Type `BACK_PAS MISC 05-01-90` (if today's date were May 1, 1990).
5) Copy the new executable to the appropriate MEMMAN directory, overwriting the old version.
   a) Change to the source file directory if needed (D:\TP\DEVELOP)
   b) Type `COPY MAN_MAIN.EXE C:\SEQ-TOOL\MEMMAN\MEMMAN.EXE`
   c) Type `DEL MAN_MAIN.EXE`

Appendix A: Batch Files

A.1 Turbo

```
d:
cd \tp
turbo
```

A.2 Pack

```
echo off
if "==%1" goto BAD
:GOOD
C:\ARC\PKARC U %1 %2 %3 %4 %5 %6 %7 %8 %9
GOTO END
:BAD
echo ------------------------------
echo Useage: PACK filename.ARC *.*
echo ------------------------------
:END
```

A.3 UnPack

```
echo off
if "==%1" goto BAD
:GOOD
C:\ARC\PKXARC %1 %2 %3 %4 %5 %6 %7 %8 %9
GOTO END
:BAD
echo ------------------------------
echo Useage: UNPACK filename.ARC *.*
echo ------------------------------
:END
```

A.4 List

```
echo off
if "==%1" goto BAD
:GOOD
c:\ARC\PKXARC -v %1
GOTO END
:BAD
echo -------------------------
echo Useage: LIST filename.ARC
echo -------------------------
:END
```

A.5 Delete

```
C:\ARC\PKArc d %1 %2 %3 %4 %5 %6 %7 %8 %9
```

A.6 Clean

```
Echo off
If "==%1" GOTO :Syntax

If %1==MAN GOTO :MAN
If %1==man GOTO :MAN
If %1==MISC GOTO :MISC
If %1==misc GOTO :MISC
GOTO :Syntax

:MAN
c:\arc\pkarc -u MAN_PAS MAN_????.PAS MEMMAN.*
del memman.*
del man_????.pas
del man_????.tpu
del man_????.bak
GOTO :GoodEnd

:MISC
c:\arc\pkarc  -u  MISC_PAS  GW_?????.PAS  WINDOWS.PAS  MOUSE.PAS
AUX?????.PAS COLORS.PAS COMWATCH.PAS EM_MGR.PAS
del gw_?????.*
del windows.*
del mouse.*
del aux?????.*
del colors.*
del comwatch.*
del em_mgr.*
GOTO :GoodEnd

:Syntax
Echo Command Usage:
Echo.
Echo CLEAN MAN or CLEAN MISC
Echo.
Echo Packs and cleans source code files for MAN_PAS.ARC
Echo or MISC_PAS.ARC
Echo.
GOTO :BadEnd

:GoodEnd
Echo Cleaning completed.
:BadEnd
```

A.7 Back_Pas

```
Echo off
If "==%1" GOTO :Syntax
If "==%2" GOTO :Syntax

If %1==MAN GOTO :Copy
If %1==man GOTO :Copy
If %1==MISC GOTO :Copy
IF %1==misc GOTO :Copy
GOTO :Syntax

:Copy
mkdir a:\%2
copy d:\tp\develop\%1_pas.arc a:\%2\%1_pas.arc
GOTO :GoodEnd

:Syntax
Echo Command Usage:
Echo.
Echo BACK_PAS MAN MM-DD-YY or BACK_PAS MISC MM-DD-YY
Echo.
Echo Backs-up PASCAL archive files to A:\MM-DD-YY\*.*
Echo.
GOTO :BadEnd

:GoodEnd
ChkDsk A:
Echo Backup complete.
:BadEnd
```

Appendix B: More About PKARC & PKXARC

The following information was taken from the PKARC program documentation file (PKARC.DOC).

B.1 General Information

PKARC will run on any IBM PC/XT/AT/RT/jr/Portable/Convertible or any MS-DOS compatible computer running PC/MS-DOS 2.0 or higher with a minimum of 128K free RAM.

IBM is a registered trademark of the International Business Machine Corporation. MS-DOS is a registered trademark of Microsoft Inc.

If you have any questions or comments about PKARC send them to Phil Katz at:

RBBS-PC of FARGO, Loren Jones SYSOP
Fargo, North Dakota
701-293-5973
300/1200/2400 baud, 24 hours a day

or

Exec-PC IBM BBS, Bob Mahoney SYSOP
Shorewood, Wisconsin
414-964-5160
300/1200/2400 baud, 24 hours a day

Special thanks to Loren Jones, Bob Mahoney, Alan Losoff, Gene Alm, Mike Shawaluk, Paul Waldinger, Arny Krueger, Mark Tellier, Joe Vincent, David Wyatt, and everyone else who has contributed to the PKARC effort.

B.2 License

You are free to use, copy and distribute PKARC for noncommercial use IF:

NO FEE IS CHARGED FOR USE, COPYING OR DISTRIBUTION.

IT IS NOT MODIFIED IN ANY WAY.

Clubs and user groups may charge a nominal fee (less than $10) for expenses and handling while distributing PKARC.

Volume discounts, site licenses, commercial licenses and custom versions of PKARC and PKXARC are available. Write to the address below for more information.

This program is provided AS IS without any warranty, expressed or implied, including but not limited to fitness for a particular purpose.

If you find PKARC fast, easy, and convenient to use, a contribution of $20 would be appreciated. With each contribution of $45 or more you will be registered to receive a diskette with the next version of PKARC and PKXARC when available. Please state the current versions of PKARC and PKXARC that you have.

Send contributions to:
PKWARE, Inc.
7032 Ardara Avenue
Glendale, WI 53209

Appendix C: Turbo Pascal Settings

Any TP pull down menus with optional settings are listed below, with pertinent options following. Irrelevant or non-modifiable options are not listed.

FILE				
	Change Directory			D:\TP\DEVELOP
COMPILE				
	Destination			Disk
	Primary File			MAN_MAIN.PAS
OPTIONS				
	Compiler			
		Range Checking		Off
		Stack Checking		On
		I/O Checking		On
		Force far calls		Off
		Overlays allowed		Off
		Align data		Word
		Var-string checking		Relaxed
		Boolean evaluation		Short Circuit
		Numeric processing		8087/80287
		Emulation		On
		Debug information		On
		Local symbols		On
		Memory Sizes		
			Stack heap	16384
			Low heap limit	0
			High heap limit	655360
	Linker			
		Map file		Off
		Link buffer		Disk
	Environment			
		Config auto save		Off
		Edit auto save		On
		Backup files		On
		Tab size		8
		Zoom windows		On
	Directories			
		Turbo directory:		D:\TP
		EXE & TPU directory:		D:\TP\DEVELOP
		Include directories:		D:\TP\DEVELOP
		Unit directories:		D:\TP\DEVELOP
DEBUG				
		Integrated debugging		On
		Standalone debugging		Off
		Display swapping		Smart

Appendix D: Glossary

ASCII	American Standard Code for Information Interchange
VIM	Voyager Interstellar Mission
DOS	Disk Operating System
MSDOS	Microsoft Disk Operating System
PKARC	MSDOS program used to place files into an archive
PKXARC	MSDOS program used to extract files from an archive
TP	Turbo Pascal
TPU	Turbo Pascal Unit

Appendix E: Notes

1. Jet Propulsion Laboratory, "MemMan Functional User's Guide"

2. Borland International, "Turbo Pascal 4.0 Owner's Handbook", Copyright 1987

3. IBM, "Disk Operating System Version 3.30 Reference", 1987