ZCPR2 -- An Enhanced Z80-Based Replacement for the CP/M CCP

Installation Instructions for ZCPR2

1.0 Introduction

Installation of ZCPR2 is a somewhat involved process, and the installer should have a fair-to-good working knowledge of the following:

```
        1) 8080 and Z80 assembly language programming
        2) CP/M 2.2
        3) the CP/M SYSGEN procedure and how to replace the
           operating system on the system tracks of the
           target system
```

The memory images of a CP/M system and a ZCPR2-based system are presented below for a conventional CP/M system. The actual addresses may vary from system to system, and the installer should be aware of what these addresses are for the specific target system.

```
                         -- Diagram 1 --

Address         CP/M Image               ZCPR2 Image


                --------------------     ----------------------
                | BIOS              |    | BIOS with Modified |
                |                   |    |    Cold Boot *      |
BDOS+0E00H-->   --------------------     ----------------------
                | BDOS              |    | BDOS (No Change)   |
CCP +0800H-->   --------------------     ----------------------
                | CP/M 2.2 CCP      |    | ZCPR2 *             |
BOOT+0080H-->   --------------------     ----------------------
                | BOOT              |    | BOOT               |
BASE+xxxxH-->   --------------------     ----------------------
                | Junk and SYSGEN   |    | Junk and SYSGEN    |
BASE= 100H-->   --------------------     ----------------------
```

            CP/M and ZCPR2-based SYSGEN Memory Images

Installation requires a modified BIOS image and a ZCPR2 image to be placed over the original CP/M 2.2 BIOS and CCP images. The rest of the system can stay the same. Those new images are marked with an asterisk (*) above.

            Typical address values are indicated below:

Value	SYSGEN Image Conventional CP/M	SYSGEN Image Morrow CP/M
xxxxH	800H	~ 980H
BOOT = BASE + xxxxH	900H	1080H
ZCPR2 = BOOT + 80H	980H	1100H
BDOS = ZCPR2 + 800H	1180H	1900H
BIOS = BDOS + 0E00H	1F80H	2700H
End of Operating System	????H	2DFFH

            Refer back to Diagram 1 for clarification.

ZCPR2 -- An Enhanced Z80-Based Replacement for the CP/M CCP

The installation process involves three steps:

        1) Configuring your BIOS and setting the ZCPR2 options.
        2) Overlaying the CCP with ZCPR2 and the old BIOS with
           the new BIOS in the SYSGEN Image.
        3) Placing the new SYSGEN Image onto the Operating
           System tracks of the disk.

The installer is invited to refer to the CP/M 2.0 Alteration Guide
by Digital Research for details in procedure. This manual
specifically covers installation of a BOOT and BIOS, but these
concepts can be logically extended to the CCP (ZCPR2) as well.

1.1 Required Software for Installation

ZCPR2 is to be installed on a working CP/M 2.2 system. The specific
commercial software required to do this installation is:

        1) A working CP/M 2.2 System
        2) the MAC assembler of Digital Research
           (another assembler may be substituted, but source
           code modification to ZCPR2 may be required to do
           an installation with an assembler other than MAC)
        3) a debugger, like DDT, for the overlay process
        4) a disk utility, like SYSGEN, to be used to place the
           operating system image onto the operating system
           tracks on disk

The non-commercial software required for ZCPR2 installation is:

        1) source to ZCPR2

        2) the CCPLOC program (may be required, and is provided
           in the ZCPR2 distribution)

        3) source to the BIOS of the target machine or the
           overlay for the BIOS Cold Boot routine for the
           target machine (this may be available commercially)

        4) the ZCPR2 utilities (source is typically not re-
           quired for installation since the GENINS program
           can perform installation in most cases); these
           utilities and their sources an included in the
           ZCPR2 distribution

A listing of the ZCPR2 distribution software, with CRC values so
that you can check to see that you have good copies of everything,
is included in Appendix A of this document.

```
1.2  Required Hardware for Installation and Running ZCPR2
```

The hardware requirements for the installation of ZCPR2 are as
follows:

```
        CP/M 2.2 - based system (or ZCPR2 - based system)
        8080 or Z80 microprocessor
        32K bytes of memory
        110K bytes of disk space for source, BAK, and HEX files
        computer terminal
```

```
    The hardware requirements for running ZCPR2 are:
```

```
        ZCPR2 - based system
        Z80 microprocessor
        48K bytes of memory
        110K bytes of disk space (recommended minimum)
        computer terminal
```

```
2.0 Installing ZCPR2
```

In order to install ZCPR2 on a target microcomputer, the user must
know two basic things about his Console Command Processor (CCP):

```
        1) Where it is located when it is executed
        2) Where it is located in the SYSGEN Image or, for
           systems which don't support SYSGEN, where his CCP
           is located on disk and how to place the new ZCPR2
           on top of it
```

```
    Note Diagrams 1 and 2:
```

```
                        -- Diagram 1 --

Address         CP/M Image               ZCPR2 Image


                --------------------      ---------------------
                | BIOS             |      | BIOS with Modified |
                |                  |      |   Cold Boot *      |
BDOS+0E00H-->   --------------------      ---------------------
                | BDOS             |      | BDOS (No Change)   |
CCP +0800H-->   --------------------      ---------------------
                | CP/M 2.2 CCP     |      | ZCPR2 *            |
BOOT+0080H-->   --------------------      ---------------------
                | BOOT             |      | BOOT               |
BASE+xxxxH-->   --------------------      ---------------------
                | Junk and SYSGEN  |      | Junk and SYSGEN    |
BASE= 100H-->   --------------------      ---------------------

        CP/M and ZCPR2-based SYSGEN Memory Images
```

```
                      -- Diagram 2 --

Address           CP/M Image              ZCPR2 Image


                  --------------------    ----------------------
                  | BIOS             |    | BIOS with Modified |
                  |                  |    |    Cold Boot       |
BDOS+0E00H-->     --------------------    ----------------------
                  | BDOS             |    | BDOS (No Change)   |
CCP +0800H-->     --------------------    ----------------------
                  | CP/M 2.2 CCP     |    | ZCPR2              |
CCP Base  -->     --------------------    ----------------------
                  | Scratch Area     |    | Scratch Area       |
100H      -->     --------------------    ----------------------
                  | CP/M Buffers et al|   | ZCPR2 Buffers et al|
   0H     -->     --------------------    ----------------------


          CP/M and ZCPR2-based Executing Images
```

Question 1: Where is the CCP located when it is executed?

A program called CCPLOC is provided in the ZCPR2 distribution.
This program, whose COM file is set up to run starting at 100H, will
return to you the base addresses of your CCP, BDOS, and BIOS. This
program by itself answers the first question for you.

If you are running a non-ORG 0 CP/M system, you will have to
reassemble CCPLOC in order to run it. This discussion is oriented
towards ORG 0 CP/M systems, but it is not difficult to adjust
offsets for non-ORG 0 systems.

Question 2: Where is the CCP located in the SYSGEN Image or, for
systems which don't support SYSGEN, where is the CCP located on disk
and how will the new ZCPR2 be placed on top of it?

This question is not answered nearly so easily. If you have the
ability to SYSGEN your system, it is much easier (commonly) than if
you do not. You must, after assembling ZCPR2 properly, integrate it
into the SYSGEN (or disk) image of CP/M. This can be done by
obtaining a SYSGEN image of your system, scanning it via a debugger
such as DDT to find the offset for the CCP, reading ZCPR2 in on top
of the CP/M 2.2 CCP, and finally running SYSGEN again to place the
resultant system on disk. If you DO NOT have SYSGEN capability, a
Disk Utility program is required to locate the CCP on disk and then
write ZCPR2 on top of the old one. The net result of this
integration is the placement of ZCPR2 onto disk in the proper place
so that it will be loaded with the rest of CP/M on cold boot and
executed properly.

To find the original CP/M 2.2 CCP, you typically have to locate it
by its appearance. It may be stored contiguously on disk, so, once
it is found, a sequential overwrite is all that is required. This
is not always the case, however, and your particular situation may
be different. Probability is extremely high that it is stored
contiguously in the SYSGEN image.

The CCP starts with two and ONLY TWO jump instructions followed by a
buffer area (possibly containing an initial command and/or the
Digital Research copyright notice).   The Digital Research manuals
show the CCP to reside at address 980H in the SYSGEN Image, but this
may vary with the particular system (note previous diagrams).   To
find this image, use DDT or some other debugger, load the SYSGEN
image, and examine memory starting at around 900H for the two (and
ONLY two) jumps described above. If you find an area with more than
two jumps (a group of them), you are probably looking at the BIOS
and should go lower for the CCP. The CCP will probably start on an
even page or half-page address (like 900H, 980H, 1100H, etc).

Now the locations of the CCP in the SYSGEN image and in the running
image are known.   Record these addresses for later.   You are now
ready for the integration of ZCPR2 and the new BIOS into your
system. To do this, perform the following steps:

1)   Assuming that you can use SYSGEN, obtain a SYSGEN image of your
     current CP/M system and save it on disk.   Also run CCPLOC and
     obtain its values.

2)   Edit ZCPRHDR.LIB and set the CPRLOC equate to the address of the
     running CCP image obtained above.   Set any flags and values as
     you desire (see the ZCPR2 Customization section).

3)   Edit your BIOS or BIOS Overlay and set up your Cold Boot routine
     to perform the necessary initializations for the ZCPR2 features
     you have selected (see the section on Modification of the BIOS
     Cold Boot Routine below).   When satisfied, end the edit session.

4)   Assemble ZCPR2 with MAC (or   equivalent).   This assembler is
     required because of the MACROs used.   Only the resultant HEX
     file is required for the rest of the installation procedure, so
     you can get rid of ZCPR2 source if you are running short on disk
     space.

5)   Assemble your BIOS or BIOS Overlay.   Again, only the resultant
     HEX file is required, so you can get rid of the source once the
     assembly is complete.

6)   Load the SYSGEN image into memory with DDT (or equivalent).
     Once loaded, verify that the original CCP is at the IMAGE
     address found above and compute the integration offset using the
     DDT H command:

                    H<IMAGE address>,<RUNNING address>

     The second number displayed gives you the OFFSET value required
     for steps 7 and 8.

7)   Integrate ZCPR2 into your SYSGEN image via DDT's I and ROFFSET
     commands.   Use IZCPR2.HEX (or the name of your version of
     ZCPR2) to load the FCB and ROFFSET (where OFFSET was computed in
     Step 6) to load the ZCPR2.HEX file into memory at the proper
     location.   Check to see that ZCPR2 is indeed properly loaded by
     examining the SYSGEN IMAGE area.

```
8)  Integrate your new ZCPR2 BIOS into your SYSGEN image via DDT's I
    and ROFFSET commands.   Use IBIOS.HEX (or the name of your
    version of the BIOS) to load the FCB and ROFFSET (where OFFSET
    was computed in Step 6) to load the BIOS.HEX file into memory at
    the proper location.   Check to see that the BIOS is indeed
    properly loaded by examining the SYSGEN IMAGE area.

9)  Place the new system on disk by running SYSGEN and NOT loading
    the system from disk (use the memory image).
```

2.1 ZCPR2 Installation Example

For further clarification of the above procedure, the following is a
sample terminal session which outlines the steps taken. It is run
from a ZCPR2 system, and the screen displays are edited to omit
unnecessary detail and improve clarity.

```
B1>;
B1>;  Sample terminal session which illustrates ZCPR2 installation
B1>;   This is done from a running ZCPR2 system and shows some of
B1>; the extended features of ZCPR2
B1>;
B1>
B1>;
B1>;  Step 1a -- SYSGEN
B1>;
B1>hdsysgen
SYSGEN VER 2.2
SOURCE DRIVE NAME (OR RETURN TO SKIP)a
SOURCE ON A, THEN TYPE RETURN
FUNCTION COMPLETE
DESTINATION DRIVE NAME (OR RETURN TO REBOOT)

B1>save 45 cpzm.cpm
B1>;
B1>;  Step 1b -- Run CCPLOC to get addresses
B1>;
B1>ccploc
CCPLOC  Version 1.0

CP/M Element    Base Page       Base Address
------------    ---------       ------------
    CCP           BDH             BD00H
    BDOS          C5H             C506H
    BIOS          D3H             D300H
```

```
B1>;
B1>;  Note the files I am working with
B1>;
B1>xdir

XDIR III, Version 1.3    Vertical Listing by File Type and Name
 Disk: B  User:   1, File Attributes:  Non-System

Filename.Typ Size K    Filename.Typ Size K    Filename.Typ Size K
-------- --- ------    -------- --- ------    -------- --- ------
CHBIOSZ .ASM    52 CPZM    .CPM    12  ZCPRHDR .LIB    12
ZCPR    .ASM    56 CBIOSHDR.LIB    12
     5 Files Occupying  144K,    33 Files on Disk and  4360K Free


B1>


B1>;
B1>;  Step 2 -- Edit ZCPRHDR.LIB and set all options and values
B1>;
B1>ed zcprhdr.lib
     -- Detail Left Out --
B1>
B1>;
B1>;  Step 3 -- Edit CHBIOSZ.ASM and CBIOSHDR.LIB and set Cold
B1>;      Boot Inits
B1>;      You will probably only have to deal with one BIOS file
B1>;
B1>ed chbiosz.asm
     -- Detail Left Out --
B1>ed cbioshdr.lib
     -- Detail Left Out --
B1>;  I'll erase the backup files now
B1>era *.bak
CHBIOSZ .BAK  |  ZCPRHDR .BAK  |  CBIOSHDR.BAK
B1>
B1>;
B1>;  Step 4 -- Assemble ZCPR2
B1>;      I will also run TIME to show you how long it takes
B1>;
B1>time;mac zcpr2 $pz sz;time
TIME  Version 1.1    1:43:38 AM on Thursday December 16, 1982
CP/M MACRO ASSEM 2.0
C4FC               <-- Note the ZCPR2 MUST end before BDOS
01CH USE FACTOR    <-- begins!
END OF ASSEMBLY

TIME  Version 1.1    1:45:18 AM on Thursday December 16, 1982
B1>
```

```
B1>;
B1>;  Step 5 -- Assemble BIOS
B1>;
B1>time;mac chbiosz $pz sz;time
TIME  Version 1.1    1:49:15 AM on Thursday December 16, 1982
CP/M MACRO ASSEM 2.0
FCE0
01AH USE FACTOR
END OF ASSEMBLY

TIME  Version 1.1    1:50:32 AM on Thursday December 16, 1982
B1>xdir

XDIR III, Version 1.3    Vertical Listing by File Type and Name
 Disk: B  User:  1, File Attributes:  Non-System

Filename.Typ Size K   Filename.Typ Size K   Filename.Typ Size K
-------- --- ------   -------- --- ------   -------- --- ------
CHBIOSZ .ASM    52   CHBIOSZ .HEX     8   CBIOSHDR.LIB    12
ZCPR2   .ASM    56   ZCPR2   .HEX     8   ZCPRHDR .LIB    12
CPZM    .CPM    12
    7 Files Occupying  160K,    35 Files on Disk and  4344K Free

B1>;
B1>;  Step 6 -- Run DDT and locate the CCP
B1>;
B1>ddt cpzm.cpm
DDT VERS 2.0
NEXT  PC
2E00 0100
-d900,90f       <-- Start looking around here
0900 31 80 E7 3E 06 3C 3C FE 1B CA 00 C2 DA 11 E7 D6
-da00,a0f
0A00 31 00 01 01 01 0C C5 CD 0F E4 21 00 BE 11 00 04
-db00,b0f
0B00 31 00 01 01 01 11 C5 CD 0F E4 21 00 C0 11 00 02
-db80,b8f
0B80 31 00 01 01 09 01 CD A8 00 21 00 D2 11 00 C2 0E
        -- Detail Left Out --
-d1100          <-- I found it at 1100H; note the 2 JMP's
1100 C3 FF BD C3 FB BD 50 10 20 20 20 20 20 20 20 20
1110 20 20 20 20 20 20 20 20 00 00 00 00 00 00 00 00
1120 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
        -- Detail Left Out --
-h1100,bd00
CE00 5400      <-- Integration Offset is 5400 for my system
-^C            <-- Return to CP/M; I know that CPRLOC will be
                   BD00H and the IMAGE offset is 1100H
```

```
B1>
B1>;
B1>;  Steps 7 and 8 -- Perform integration of ZCPR2 and BIOS
B1>;
B1>ddt cpzm.cpm           <-- Edit CP/M Image
DDT VERS 2.0
NEXT  PC
2E00 0100
-h1100,800               <-- Compute End Address of CCP
1900 0900                <-- (CCP is 800H bytes long)
-f1100 18ff 0            <-- Zero Fill Area (I like to do this)
-izcpr2.hex              <-- Overlay ZCPR2
-r5400                   <-- Offset Computed Above
NEXT  PC
2E00 0000
-h1900,e00               <-- Compute Beginning of BIOS
2700 0B00
-f2700 2dff 0            <-- Zero Fill Area (I like)
-ichbiosz.hex            <-- Overlay BIOS
-r5400                   <-- Offset Computed Above
NEXT  PC
2E00 0000
-^C                      <-- Done


B1>;
B1>;  Step 9 -- Perform SYSGEN
B1>;
B1>save 2dh cpzm.cpm          <-- Save file (note HEX page count)
Erase CPZM    .CPM?y
B1>hdsysgen                   <-- SYSGEN to disk from memory
SYSGEN VER 2.2
SOURCE DRIVE NAME (OR RETURN TO SKIP)            <-- Skip
DESTINATION DRIVE NAME (OR RETURN TO REBOOT)a    <-- Select A
DESTINATION ON A, THEN TYPE RETURN
FUNCTION COMPLETE
DESTINATION DRIVE NAME (OR RETURN TO REBOOT)     <-- REBOOT!


B1>
```

3.0 ZCPR2 Customization

The file ZCPRHDR.LIB contains all of the customization equates and
values for ZCPR2, and it is typically not necessary to edit
ZCPR2.ASM in order to configure it to your desires. Editing
ZCPRHDR.LIB is the only procedure necessary to perform such
customization.

In this section of the Installation Manual I will go through each of
the equates and value settings in ZCPRHDR.LIB in sequence (from
beginning to end). This allows you to simply step through the file
with an editor while you are reading this manual. The
documentation in this manual augments the documentation in
ZCPRHDR.LIB to give you a fairly complete explanation of each
option.

It is important that you have read the Concepts Manual before
performing this customization. Knowledge of how CP/M 2.2 works and
of the contents of the Concepts Manual is assumed in this section of
the Installation Manual.

   1.   REL -- Configures CPRLOC (CPRLOC equ 0) for integration via
        MOVCPM rather than the DDT/SYSGEN technique outlined above;
        set to TRUE for MOVCPM integration or FALSE for DDT/SYSGEN
        integration.

   2.   BASE -- Base address of your CP/M system; standard CP/M has a
        base of 0, but some CP/M systems (such as for the TRS-80
        Model I) start physical RAM memory at a higher address;
        equate BASE to the starting RAM memory address of your
        system.

   3.   CPRLOC -- This is the starting address of ZCPR2; set the
        second CPRLOC equate to the address you obtain from CCPLOC;
        if you have a "standard" system, you may wish to use the
        MSIZE, BIOSEX, and CPRLOC equates which are commented out
        instead of the simple CPRLOC I prefer to use.

   4.   CPRR -- This is the value of OFFSET, and you may set this
        equate according to your system as desired; it is not used by
        ZCPR2, but it does provide you with the required value
        without having to compute it via the H command in DDT.

   5.   BIOS -- This is a standard equate and usually does not need
        to be changed.

   6.   COMTYP and SUBTYP -- These MACROs allow you to define the
        file type of the COM and SUB files; this is an option which
        improves flexibility, and, in some environments, it may be
        desirable to type COM files as something else, like OBJ.

   7.   SUBON -- This turns on the SUBMIT file processing facility
        of ZCPR2.

   8.   DRVPREFIX -- This enables or disables the ability to specify
        commands beginning with a disk/user prefix, like "A4:X" or
        "B:RUNIT".

A side effect of DRVPREFIX being TRUE is that the current
disk/user is automatically scanned before the path search
begins.

9. DRVPFATT   -- This   flag allows the user to select the
    attributes of the files in the current disk/user to be
    examined if the DRVPREFIX flag is TRUE.   The following values
    are defined:

             DRVPFATT Files Selected
             -------- --------------
                  0   System
                 80H   Directory
                  1   Both System and Directory

10. USERON -- This flag enables or disables the ability of the
    user to select different user areas via the DU: prefix.

11. xxxxON -- These flags enable or disable various ZCPR2-
    resident commands.   MACROs are available later in
    ZCPRHDR.LIB to define the names of the commands if the user
    wishes to change them.   These xxxxON flags are:

             Flag   Commands Affected
             ----   -----------------
             DIRON   DIR
             LTON   LIST, TYPE
             GOON   GO
             ERAON   ERA
             SAVEON   SAVE
             RENON   REN
             GETON   GET
             JUMPON   JUMP

12. INTPATH -- This flag determines if ZCPR2 uses an internal
    path or an external path for COM file searches.   If INTPATH
    is FALSE, then External Path is selected and the next equate,
    PATH, defines the base address of this External Path.   If
    INTPATH is TRUE, then the following IPATH macro is used to
    define the Internal Path, and this path is located
    within ZCPR2 itself (taking up space).

13. ERAxx and ERDFLG -- These equates deal with the ERA command
    specifically, and are simply ignored if ERA is turned off.
    They work together to provide the following features to the
    ERA command:

        -- Settings --
        ERAOK   ERAV   Feature
        FALSE   anything   ERA will not ask to verify before
                                performing the erase
        TRUE   FALSE   ERA will always ask to verify
                                before performing the erase
        TRUE   TRUE   ERA will ask to verify before
                                performing the erase only if the
                                ERDFLG (recommended to be "V")
                                flag appears in the command line
                                after the file spec

14. DUPRMPT -- This equate enables or disables the "du>" prompt. If enabled, the prompt for user input will appear as "d>" or "du>" (see the SUPRES flag below also). If disabled, the prompt will only appear as ">".   See also CPRMPT and SPRMPT below.

15. MULTCMD, et al -- This equate enables or disables the ZCPR2 multiple command line feature. If set to TRUE, the feature is enabled and the CMDSEP, CLBASE, and BUFLEN equates take effect.

    These other equates mean the following:

    CMDSEP -- define the character used to separate each entry in the command line; if changed from the default of ";", all the ZCPR2 utilities have to be reinstalled

    CLBASE -- base address of the Multiple Command Line Buffer

    BUFLEN -- size of the Multiple Command Line Buffer in bytes; MUST be less than 254 (I have found 200 to be quite sufficient)

16. Wheel Byte Selection -- The Wheel Byte facility of ZCPR2 is used to make ZCPR2 a more secure system by enabling and disabling selected ZCPR2 resident commands.   With this facility enabled, a Wheel Byte, which is simply a byte whose value is 0 or 0FFH that exists somewhere in memory, is examined before a wheel-based command is executed. If this byte is not zero, then the command executes; if it is zero, then the command is not allowed to execute and is exited with an error message. The ZCPR2 transient WHEEL is employed by a privileged user to set and reset the Wheel Byte; with the WHEEL command, a password is required before any action is taken.

    The next set of equates is used to (1) define the address of the Wheel Byte and (2) select the commands which are to respond to the Wheel Byte setting and abort with error if the Wheel Byte is a zero.

        WHLADR - Address of Wheel Byte

        WERA, WREN, WLT, WGO, WSAVE, WGET, WJUMP, WDU –
            If TRUE, these equates make their associated
            commands respond to the Wheel Byte:

Equate	Association	Equate	Association
WERA	ERA	WSAVE	SAVE
WREN	REN	WGET	GET
WLT	LIST/TYPE	WJUMP	JUMP
WGO	GO	WDU	DU: Change

17. The next equates are used to define the stack used by ZCPR2. The stack may be placed internal to ZCPR2 (taking up space) or external to ZCPR2. The INTSTACK equate determines if the stack is internal to ZCPR2; if TRUE, such is the case. If INTSTACK is FALSE, then the STKBASE equate determines where

the bottom of the 48-byte stack is located in memory. The
installer should not change the STACK equate following
STKBASE since 48 bytes (and no more) are equired for the
stack. Note that the stack grows DOWN to STKBASE. Summary:

```
INTSTACK - TRUE if stack is internal to ZCPR2
STKBASE  - if INTSTACK is FALSE, then this is the
                 address of the bottom of the stack
```

18. Simple Customization Symbols -- These symbols provide for
     further customization of ZCPR2 to a user's particular tastes
     and hardware facilities. They are covered briefly here since
     they are all quite straight-forward.

Option Name	Function
NLINES	Number of lines on the user's CRT for paging
WIDE	This equate is used to select a narrow or wide display under the DIR command; if WIDE is equated to TRUE, each file name is separated by two spaces, a FENCE, and two more spaces; if WIDE is equated to FALSE, each file name is separated by one space, a FENCE, and one more space
FENCE	This equate defines the character used to separate entries in a display caused by the DIR and ERA commands
PGDFLT	This is the Paging Default flag for the TYPE command; if PGDFLT is set to TRUE, the TYPE com- mand will page its output by default and the P option on the TYPE command (see below) will pro- hibit paging; if PGDFLT is set to FALSE, the TYPE command will NOT page its output by default and the P option will enable paging
PGDFLG	This sets the option character in the command line for the TYPE command (the 'P' mentioned above); if the user wishes to change this option character, he need only change this equate
MAXUSR	This is the largest user number recognized by the user number processor; if the user wishes to pro- tect the higher user areas, he may set this symbol to the highest area normally accessable; 31 is the largest permitted value for MAXUSR; in any case, 15 is the largest user number one may log into (unless MAXUSR is less than 15), but it is not the largest user number that may be referenced by DIR, ERA, and the other ZCPR2-resident commands (like, "DIR A22:")
MAXDISK	This equate defines the number of logical disks the user supports on his system; it may not be greater than 16
SYSFLG	This is the option character for the DIR command

line which is used to specify that DIR search All Files (both $SYS and $DIR) for its display; the distributed default for this is 'A'

SOFLG This is the option character for the DIR command line which is used to specify that DIR search ONLY the $SYS files for its display; the distributed default for this is 'S'

SUPRES Set SUPRES to TRUE to suppress printing the user number when the user is under User Number 0 or set SUPRES to FALSE to ALWAYS display the User Number with the ZCPR2 prompt; with SUPRES set to TRUE, a user on B: in user 0 sees 'B>' as the prompt, but with SUPRES set to FALSE, a user on B: in user 0 sees 'B0>' as the prompt

SPRMPT This is the ZCPR2 prompt character which indicates that a SUBMIT file is in execution; by default it is set to '$', so prompts like 'A$' appear during SUBMIT file execution

CPRMPT This is the ZCPR2 prompt character which indicates that ZCPR2 is awaiting a user console command; by default it is set to '>', so prompts like 'A>' appear during user input to ZCPR2; note that I have set the MSB of this character so that it will probably be unique to all characters sent to the console -- the reason for this is so that the ZEX utility will have something unique to look for in order to resume processing from user input -- for more details, refer to the documentation on ZEX

NUMBASE This is the escape character used by those commands which require a DECIMAL number as an argument; placing this character after the number argument switches the base to HEXADECIMAL; for example, 'SAVE 15 MYFILE' can be expressed as 'SAVE FH MYFILE' if NUMBASE is set to 'H' (the default)

SECTFLG This character constant is the suffix option for the SAVE command which specifies that sectors, as opposed to pages, are to be saved; the default value is 'S'

CURIND This character defines the symbol used to represent the current disk or current user in the command search path; its recommended value is '$'

COMMENT This character is used to make the entire ZCPR2 command line into a comment if it appears as the first character in the line; its recommended value is ';'

19. NCHARS and CTABLE -- NCHARS is used to specify the number of characters in a ZCPR2-resident command. Its recommended value is 4 (to limit the overall size of the commands). C-TABLE is a table containing the text of each command name

(each text entry must be NCHARS long) and the address of each command routine. The table is structured as follows:

```
            DB    'TEXT'    ;Command Text
            DW    ADR       ;Command Address
            ...             ;And So On ...
```

With this table you can change the name of any ZCPR2-resident command to anything you want, within reason.   'TYPE' may be changed to 'PRNT', for example. As a rule, do not change the DW entries without a very good reason.   These symbols are fixed within ZCPR2.ASM.

20. EXTFCB and FCBADR -- EXTFCB indicates where the FCB used to contain intermediate and command file names is to be located. If EXTFCB is TRUE, this FCB is located external to ZCPR2, and its address is indicated by FCBADR.   If EXTFCB is FALSE, this FCB is located internal to ZCPR2, and more space is taken up within ZCPR2 (36 bytes more) than would be taken up if EXTFCB was TRUE.

21. CMDRUN, ROOTONLY, and CMDFCB -- CMDRUN enables the ability of ZCPR2 to execute an extended command processor should the search for a COM file along the command search path fail. If CMDRUN is TRUE, this facility is enabled.   Once it is TRUE, ROOTONLY and CMDFCB are active.

ROOTONLY indicates whether ZCPR2 will search along the path for the extended command processor or jump directly to the root (last irectory in the path). If ROOTONLY is FALSE, the path search is done. ROOTONLY being FALSE causes the execution of the extended command processor to take just a little bit more time than it would otherwise, but more flexibility is added to the system.

CMDFCB is a MACRO which defines the first 12 bytes of the FCB which defines the name of the extended command processor to be searched for. You may name it anything you desire, but it is recommended that this name contain at least 6 characters so that command prefixing may be done easily if so desired for further chaining.

Refer to the section on the Extended Command Processor in the Concepts Manual for more information.

3.1 Patching SUBMIT.COM

SUBMIT.COM may be patched to run with ZCPR2 by the following
procedure (this is recommended if the user does not have SUB.COM).
This patch simply makes it always place the $$$.SUB file on Drive
A:. Illustrative terminal session follows:


```
A>ddt b:submit.com
DDT VERS 2.0
NEXT  PC
0600 0100
-s5bb              <-- Patch is at 5BB Hex
05BB 00 1          <-- Change 0 (default drive) to 1 (drive A:)
05BC 24 .          <-- That's it!
-d5b0 5cf          <-- See change
05B0 00 00 00 00 00 00 30 30 31 20 24 01 24 24 24 20 ......001 $.$$$
05C0 20 20 20 20 53 55 42 00 00 00 1A 1A 1A 1A 1A 1A
SUB........
-^C                <-- Done
A>save 5 newsubmt.com   <-- Save new SUBMIT.COM file
```


4.0 Special Installation Considerations

4.1 ZCPR2 Sizing

Unlike the previous command processor, ZCPR 1.0, ZCPR2 sizing is
somewhat of a problem. Certain combinations of the options for the
TRUE/FALSE equated symbols can cause the size of ZCPR2 to exceed 2K
bytes, and this is typically of concern and not desired (see the
Rationale manual). If the installer, for instance, attempts to
duplicate the ZCPR configuration, then the size of ZCPR2 will exceed
2K bytes by about 30 bytes.

To help the installer to keep aware of the impact of his selection,
the following sizing chart is provided. This chart shows the
recommended option settings of the equates in question and the costs
(measured in bytes of code) associated with setting these equates.
If a cost is positive, then that says that the recommended equate
setting causes the indicated number of bytes to be consumed, and if
this setting were reversed (i.e., if currently TRUE, set it to
FALSE), then the indicated number of bytes would be saved. A
negative cost shows that the recommended setting is consuming fewer
bytes than the opposite.

ZCPR2 Sizing Chart

Symbol	Recommended Setting	Cost	Symbol	Recommended Setting	Cost
CMDRUN	TRUE	53	JUMPON	TRUE	11
DIRON	TRUE	41	LTON	TRUE	138
DRVPREFIX	TRUE	13	MULTCMD	TRUE	-67
DUPRMPT	TRUE	32	PGDFLT	TRUE	0
ERAOK	TRUE	40	RENON	TRUE	72
ERAON	TRUE	62	ROOTONLY	FALSE	0
ERAV	TRUE	19	SAVEON	TRUE	90

```
EXTFCB      TRUE        -33         SUBON       TRUE        142
GETON       TRUE        18          SUPRES      TRUE        3

GOON        TRUE        11          USERON      TRUE        3
INTPATH     FALSE       -9          WIDE        TRUE        2
INTSTACK    FALSE       -48         Wxxxx       FALSE       -8-3N*
```

        *   There are eight Wxxxx equates which enable the indicated
commands to respond to the Wheel Byte. There is an
initial overhead of eight bytes for any one equate to be
enabled and there is an additional overhead of three bytes
for each equate which is enabled. That is, to enable one
equate costs 11 bytes, and each additional equate after
that costs three bytes more.

As an installation trade-off example, if the user decides that he
does NOT want to support the SUB file capability (wishes to use ZEX
exclusively), then he can set SUBON to FALSE and save 142 bytes. If
he also decides that he does not want to have to worry about
allocating buffer space for an external FCB, he can set EXTFCB to
FALSE, but this costs him 33 bytes (33 more bytes are consumed by
having this FCB internal to ZCPR2 than having it external). Note
that due to some overlap of functions (such as RENON and SAVEON),
space savings may vary depending on the combinations of the options
selected.

4.2 Programs Which Try to Load the FCB Command Line

One thing the user and installer should be concerned with is the use
of programs which employ internal knowledge of the CP/M 2.2 CCP to
perform some of their functions. Such programs typically tamper
with the internal command line buffer of the CCP and then transfer
control to the CCP in such a way that the command line they stored
there is executed. Under ZCPR2, if MULTCMD is TRUE (the Multiple
Command Line Buffer Facility is enabled), then programs like this
will be modifying code/data area instead of the intended buffer,
which is no longer there, and frequently cause bad things to happen.

Another example is where the BIOS wishes to issue a command on Warm
or Cold Boots and it attempts to store said command in the CCP
command line buffer as described above. Similar problems will
result.

This does not mean that ZCPR2 is removing this capability, but it
does mean that the capability is not invoked in the same way. In
order to execute a desired command line set by a transient program,
for example, the transient need only store the desired command line,
terminated by a binary zero, in the proper Multiple Command Line
Buffer and set the next character pointer to the first character to
be executed. Likewise for the BIOS case.

5.0 Modifications to the BIOS Cold Boot Routines

To fully install ZCPR2 as recommended in Section 4.1, the Cold Boot
routine in the BIOS must allocate some buffer space and perform some
initializations which it does not normally do. As a working example
of what I am referring to, study the source code to the Cold Boot
Routine in my BIOS:

```
;
;  Cold Boot Routine for CHBIOSZ
;    This routine allocates and initializes buffer space as
;    required to implement ZCPR2 as recommended.
;
cboot:
        lxi     sp,tpa          ;Set up stack


;
;  Section CB-1:  Init Multiple Command Line Buffer
;
        if      multcmd         ;Multiple Commands Allowed
        lxi     d,cmdset        ;Set buffers for Multiple
                                ;  Command Processing
        lxi     h,clbase        ;Command Line Base
        call    mover           ;Copy an arbitrary 128 bytes
        endif


;
;  Section CB-2:  Init Memory-Based Named Directory
;
        if      ndir            ;Named Directory Based in Memory
        lxi     d,ndvals        ;Initial Values in Directory
        lxi     h,ndbase        ;Named Directory Base
        call    mover           ;Copy an arbitrary 128 bytes
        endif


;
;  Section CB-3:  Init IOBYTE and Redirectable I/O Drivers
;
        mvi     a,intioby       ;Initialize the I/O Byte
        sta     iobyte
        lxi     d,iodrivers     ;Default I/O Drivers
        lxi     h,iobase        ;Base Address of Drivers
        call    mover           ;Copy an arbitrary 128 bytes


;
;  Section CB-4:  Init Default Command Search Path
;
        lxi     d,path          ;Copy Cold-Boot Path
        lxi     h,pathbase      ;Into System External Path Area
        mvi     b,9             ;Always 9 bytes
        call    movlop



;
;  Section CB-5:  Print Prompt and Select User/Disk
;
        lxi     h,prompt        ;Prep for sending signon message
        call    message         ;Send the prompt
        mvi     a,0             ;Select user 0 and disk A
        sta     cpmdrv
```

```
        sta     cdisk

        jmp     gocpm

     ...

;
;  Copy An Arbitrary 128 Bytes from DE to HL
;
mover:
        mvi     b,128
movlop:
        ldax    d
        mov     m,a
        inx     h
        inx     d
        dcr     b
        jnz     movlop
        ret

     ...



;
;  Primitive Redirectable I/O Drivers which are loaded to
;  initially support I/O.  They are copied into a reserved
;  area of memory (I have allowed 1K for this) starting at
;  IOBASE (my base address is 0DC00H).  These drivers
;  provide only console I/O and disable the redirection
;  facilities.  During the course of work, these primitive
;  drivers are overlaid by the IOLOADER program with much
;  more sophisticated drivers.
;
;  Note the I use Z80 Relative Jump Instructions here because
;  this code is not executed where it is stored.  Any absolute
;  jumps would have to have their addresses adjusted after
;  the routines had been copied.
;
uart equ  0E3F9H         ;UART Address
rda  equ  4              ;UART RDA Bit

iodrivers:
     jr  ioerror         ;set no Status Routine
     db  0               ;Fill 3 bytes
     jr  ioerror         ;set no Select Routine
     db  0               ;Fill 3 bytes
     jr  ioerror         ;set no Namer Routine
     db  0               ;Fill 3 bytes

     ret                 ;No Terminal Init Necessary
     db  0,0             ;Fill 3 bytes
     jr  ustat           ;Console Input Status
     db  0               ;Fill 3 bytes
     jmp djcin           ;Console Input Char in PROM
     jmp djcout          ;Console Output Char in PROM

     jmp djcout          ;List Output Char in PROM

     jmp djcout          ;Punch Output Char in PROM
```

```
        jmp  djcin              ;Reader Input Char in PROM


        mvi  a,0ffh             ;List Status Ready
        ora  a                  ;Set Flags
        ret                     ;Done with List and No New I/O Routine

ioerror:
        xra  a                  ;No Device Assignments May be Made
        ret
ustat:
        lda  uart               ;Get status byte
        cma                     ;Inverted Logic
        ani  rda                ;RDA?
        rz
        mvi  a,0ffh             ;OK Flag
        ret



        ...

;
;  Initial Command Line in Multiple Command Line Buffer
;  Only the first five bytes are required, the fifth byte being
;  a binary 0 to indicate an empty command line, if you do not
;  wish to have an initial command
;
        if    multcmd
clbase  equ  0ff00h    ;base address of buffer
buflen  equ  200       ;size of buffer in bytes
cmdset:
        dw   clbase+4  ;pt to first char
        db   buflen    ;size of buffer
        db   0         ;don't care value
        db   'STARTUP' ;command to run on Cold Boot
        db   0         ;end of Cold Boot Command
        endif


        ...

;
;  Initial Contents of Memory-Based Named Directory Buffer
;  Note:  this defines an empty Named Directory Buffer
;
        if    ndir     ;turned on?
ndbase  equ  0fd00h    ;base address of buffer
ndvals:
        db   50         ;allow 50 entries
        db   0          ;no entries stored (entry count)
        db   0          ;dummy for no disk
        db   0          ;dummy for no user
        db   0          ;dummy for no name
        endif


        ...

;
;  Initial External Path
;  The contents of the path are outlined in the comments
;
pathbase equ  40h      ;base address of first byte of path
```

```
path:
      db    '$$'         ;current disk and current user
      db    '$',0        ;current disk and user 0
      db    1,'$'        ;disk A and current user
      db    1,0          ;disk A and user 0
      db    0            ;end of path
```

```
;
;  Note:  It is not necessary to initialize any other buffers,
;     such as EXTFCB, since their values will be initialized by
;     ZCPR2
;
```

This sample provides a fairly good model for configuration of the Cold Boot Routine in order to implement ZCPR2 as recommended. Note that four features require initialization:

        1) Multiple Command Line Buffer
        2) Memory-Based Named Directory Buffer
        3) Command Search Path
        4) Redirectable I/O Drivers

Of course, I also initialized the I/O Byte at location 3 (the STA IOBYTE instruction), an internal current disk flag, and the current user/current disk flag down at location 4 (the STA CDISK instruction).

Also, note that I just used an arbitrary transfer routine which did not consider the actual size of the significant data transferred for the initialization of the Multiple Command Line Buffer, the Memory-Based Named Directory Buffer, and the Redirectable I/O Drivers. Since both buffers and the drivers were under 128 bytes in size, this was a quick way out.

5.1 Installation Note: Multiple Command Line Buffer

     The structure of the Multiple Command Line Buffer is:

        Bytes 0 and 1: Address of Next Char in Buffer
        Byte 2: Size of Buffer
        Byte 3: Dummy used by BDOS Readln Function 10
        Bytes 4 to n: Buffer Contents, terminated by binary 0

This buffer MUST be initialized by the Cold Boot routine if the Multiple Command Line facility is enabled. The address of Byte 4 is placed in Bytes 0 and 1 (low-order first), the buffer size is placed in byte 2, and any value may be placed in Byte 3.

The command line to be executed on Cold Boot may be placed in Bytes 4 to n. Note that this line MUST be terminated by a binary 0. If you have no command line to be executed on Cold Boot, Byte 4 MUST be 0.

The Multiple Command Line Buffer plays many key roles in the ZCPR2 System. In my opinion, it is one of the most important features to implement. Through this buffer, the user can perform the "simple" function of issuing more than one command on a line, but, more importantly, programs can use it to chain to a SERIES of other programs. The MENU command uses this buffer to chain to the

selected program and then to return to itself at the proper place.
STARTUP uses it to execute a command line, and, in conjunction with
CD, STARTUP under the name of ST is used to establish a working
environment once the user has logged into a directory.   This
working environment may have a new command search path, a new named
directory structure, new device drivers, and may even be menu-
driven.

5.2 Installation Note: Memory-Based Named Directories

If the memory-based named directory facility is enabled, it **MUST** be
initialized by the Cold Boot Routine.   The structure of the named
directory buffer (on initialization) should be (assuming it is
empty):

          Byte 0:   Maximum number of entries
          Byte 1:   Number of entries stored
          Byte 2:   First Disk Number
          Byte 3:   First User Number
          Bytes 4-11: First Directory Name

If the initialized named directory buffer is to be empty, Byte 0
should be set to the number of entries permitted (based on the size
of your buffer) and bytes 1 to 4 should be set to 0. In computing
the number of entries allowed in a memory-based directory buffer,
the following simple equation applies:

          Byte 0 = (size of buffer)/10

If the remainder of this division is less than 2, subtract 1 from
Byte 0 to allow for the space taken by bytes 0 and 1. The 10 is the
number of bytes occupied by an entry (2 bytes for disk/user and 8
for directory name).

The memory-based named directory feature was created for two
reasons: to speed up the execution of a utility when it needs to
know the directory referenced by a name and to establish a "global"
directory structure.   Rather than doing a disk search, the utility
simply looks in the memory-based named directory buffer, and this
saves the time required for the disk accesses if the name is found
in the buffer.   As a "global" directory, this memory-based named
directory is always searched first, so the entries in it are
examined before the "local" directory in the NAMES.DIR file on disk
is examined.   In this way, regardless of the tree or mesh structure
imposed by logging into various environments, a set of directories
can remain common to all environments (unless, of course, ST loads a
new memory-based named directory set when the user logs into a
directory via CD).

5.3 Installation Note: Redirectable I/O System

If your ZCPR2 system is to employ the redirectable I/O system which
may be implemented by the IOLOADER and DEVICE2 commands, then the
Cold Boot Routine will have to initialize this system.   Rather than
enter a wordy explanation of what has to be done, the following is
the source to my initial Redirectable I/O Drivers as installed by
the Cold Boot Routine.   Note two things:

(1) the Cold Boot Routine simply copies these drivers into the
    buffer starting at IOBASE and
(2) all branches to routines within these drivers are relative so
    that I don't have to worry about adjusting addresses for
    their location. Source Code follows:

```
******************************************************************
*                                                                *
* Primitive I/O Drivers which are loaded at Cold Boot time.      *
*                                                                *
* Note:  DJCIN and DJCOUT are in PROM, so their addresses are    *
*        fixed and need not be adjusted because of the copy.     *
*                                                                *
******************************************************************
uart     equ     0e3f9h               ;UART address
rda      equ     4                    ;UART RDA Bit

iodrivers:
         jr      ioerror              ;no Status Routine
         db      0                    ;Fill 3 bytes
         jr      ioerror              ;no Select Routine
         db      0                    ;Fill 3 bytes
         jr      ioerror              ;no Namer Routine
         db      0                    ;Fill 3 bytes

         ret                          ;Initialize Terminal
         db      0,0                  ;Fill 3 bytes
         jr      ustat                ;Console Input Status
         db      0                    ;Fill 3 bytes
         jmp     djcin                ;Console Input Char
         jmp     djcout               ;Console Output Char

         jmp     djcout               ;List Output Char

         jmp     djcout               ;Punch Output Char

         jmp     djcin                ;Reader Input Char

         mvi     a,0ffh               ;List Status Ready
         ora     a                    ;Set Flags

         ret                          ;New I/O Driver Installation Routine

ioerror:
         xra     a                    ;No device assignments
         ret

ustat:
         lda     uart                 ;Get status byte
         cma                          ;Inverted Logic
         ani     rda                  ;RDA?
         rz
         mvi     a,0ffh               ;OK Flag
         ret
```

Aside from giving the user a lot of flexibility in the design of his
I/O system, the Redirectable I/O System also offers an alternative
to the Control-P (Console Print) function under CP/M.   Using
Control-P, the user can temporarily toggle the printer on to print

what is coming to his display.   However, under certain conditions, when the prompt reappeared, the printout would continue and require another Control-P to turn it off while under other conditions, when the prompt reappeared, the printout would be turned off by itself.

Using Redirectable I/O, a device driver for the Console which consisted of CRT Input and CRT/Printer Output could easily be created. Let's say it is named CRTPRINT and, using DEVICE, it would be turned on with a command like:

        DEVICE CON:=CRTPRINT

Turning it off would amount to issuing a command like:

        DEVICE CON:=CRT

Using such a device driver solves two problems:
   (1) the user has precise control of the printer and
   (2) the user, via the device driver, can control what goes out
        to the printer, and special character sequences which would
        normally go to the screen can be filtered from going to the
        printer as well.   Some concern had been raised about using a
        character with the most significant bit set to prompt ZEX
        from ZCPR2 since such characters place certain printers into
        a graphics mode.   Such a device driver could clear the most
        significant bit before it is sent to the printer.

5.4 Installation Note: Default Command Search Path

If the external path feature is enabled, then the Cold Boot Routine MUST initialize this path. The path is represented by a standard path expression, where "$" indicates the current user or disk, disks are numbered from 1 (disk A = 1), and users are numbered from 0. The path MUST be terminated by a binary 0.

Search paths provide WORLDS of flexibility to the user.   ZCPR2 requires at least one path to be implemented. This is the command search path, which is the path along which ZCPR2 searches for commands (COM files) when they are issued. This path may be placed internal to ZCPR2 or external to it.

The command search path is the only path REQUIRED in a ZCPR2 System, but many more paths may be established.   Each ZCPR2 System utility program is set up to deal with paths in one of two or more ways:

   (1) it may follow an external path knowing only the address of
        the beginning of the path, and
   (2) it may follow its own internal path knowing exactly what
        that path is.   The HELP program is an exception to this in
        that it follows an external path (if there is one) and, if
        that fails, follows an internal path.   The external paths
        followed by the utilities in the ZCPR2 System may be the same
        as the command search path, but they don't have to be.   ZEX,
        for instance, may be set up to ignore the command search path
        and follow a path which immediately goes to a directory
        containing a host of ZEX command files.

Configuration notes can be found in the User's Guide which provide ideas as to how paths may be established for the various ZCPR2 System utilities.   Remember that for each path established which is external to a utility program, this path must be initialized in one way or another before the utility uses it.

This initialization of external paths other than the command search path used by ZCPR2 itself may be done in at least two different ways:

  (1) the Cold Boot Routine may perform the initialization, or
  (2) PATH (which may be executed on cold boot by the STARTUP program) may be used to perform the initialization. If more than one external path is defined and the user is to be able to change the contents of this path if he wishes, then the PATH utility should be installed more than once and several versions of PATH should be created (using different names, of course).   One version of PATH is required for each external path which is to be manipulated. GENINS can be used to quickly create the different versions of PATH.

5.5 Installation Note: Sample ZCPR2 Structure

As an example, the following is an outline of the structure of my ZCPR2 system. All addresses are expressed in hexadecimal.

```
    Page 0 (Base Page):
         Location  Contents
          0 -  2   JMP to BIOS Warm Start at D300
          3         IOBYTE
          4         UDFLAG - User Number in Upper 4 Bits; Disk
                       Number (A=0) in Lower 4 Bits
          5 -  7   JMP to BDOS at C506
          8 - 37   Unused Interrupt Vectors
         38 - 3A   RST 7 used by DDT, ZSID, and others
         3B         Wheel Byte
         3C - 3F   Unused
         40 - 5B   Default Command Search Path (up to 13 elts
                       and terminating 0)
         5C - 6B   First FCB Half
         6C - 7B   Second FCB Half
         7C         CR Field if 5C-7B used as one complete FCB
         7D - 7F   Random Record Number
         80 - FF   Default DMA Buffer

     Below BD00:    Optional ZEX Monitor and Text (Variable Size)
     Address BD00:  Beginning of ZCPR2
     Address C500:  Beginning of BDOS
     Address D300:  Beginning of BIOS
     Address DC00:  Reserved Buffer Area
     Address E000:  Disk PROM
     Address E400:  Disk RAM
     Address E800 - EFFF:  Redirectable I/O Drivers (2K Avail)
     Address F000 - F600:  Free Space Optionally Used by
                              Redirectable I/O Drivers
     Address F5D0 - F5FF:  ZCPR2 Stack (48 Bytes)
     Address F600 - FCFF:  Buffer Space Used by BIOS
     Address FD00 - FEFF:  Memory-Based Named Directory Buffer
                              (512 Bytes for 50 Entries)
```

```
     Address FF00 - FFCF:  Multiple Command Line Buffer
                               (200 Bytes for text + 4 for
                                initial buffers + Dead Space)
     Address FFD0 - FFFF:  ZCPR2 External FCB
```


6.0 Installing the ZCPR2 Utilities

So far, I have been concentrating on ZCPR2 itself.   ZCPR2, however,
is only a part (less than 1/10 in terms of code size) of the ZCPR2
System. The rest of the System is made up of the ZCPR2 utilities.

Most of the ZCPR2 Utilities need to know something about the ZCPR2
environment.   In particular, these utilities need to be installed
in order to become an integral part of the overall system, and a
program, GENINS (for General Install), is provided to do this.

GENINS provides ten basic configuration options to the user.   These
are:

   0.   Named Directory Data

        a. Address of Memory-based Named Directory Buffer
           This is the starting address of the Named Directory Buffer
           stored in memory. If you do not have one, set this value
           to 0.

        b.   Number of Named Directories Allowed in a Named Directory
             File
             This value is initially set by the MKDIR program by
             creating the Named Directory File with the indicated
             number of entries.   All programs using this file must
             be set to respond to the same number of entries.   Any
             number less than 256 is allowed, and 64 is recommended
             for standardization purposes (and because it is
             reasonable as determined through experience).

        c.   Name of the Named Directory File to Scan for on Disk
             This is the name of the file to be scanned for if a
             directory name is not found in the memory-based Named
             Directory Buffer.   The recommended name (for
             standardization purposes) is NAMES.DIR.

   1.   External and Internal Path Data

        a. Enable or Disable External Paths
           The utility is told whether it may use external paths for
           its functions or not by this option.   External paths are
           used for file searches and Named Directory file searches
           primarily, but they may have other uses as well, depending
           on the utility. If external paths are available, the PATH
           command may be used to initialize them on cold boot or to
           change them dynamically during a terminal session.   If
           external paths are not to be used, then internal paths,
           unique to each utility, may be defined and employed
           instead. If neither external nor internal paths are to be
           used, then the utility (usually) will only examine the
           current directory for what it needs.   As a general rule,
           if external paths are not used, define an internal path,
```

and this internal path may be as simple as current
directory.

b.  If enabled, the Address of the External Path
    If an external path is enabled, this is the base address
    of the beginning of the path.  If external paths are
    disabled, GENINS does not give the user the option to
    set this value.

c.  Enable or Disable Internal Paths
    GENINS gives the user the option to enable and define an
    internal path regardless of his selection for external
    paths.  Some commands, HELP in particular, may use both
    paths if they are available.  In the case of HELP, when
    looking for a HELP file, it searches along the external
    path first, and, failing that, searches along its
    internal path.

d.  If enabled, define the internal path for that particular
    utility
    The internal path is defined if the user selected to
    enable internal paths.  Symbolic notation is used to
    define the path, and the user is prompted for each path
    element in sequence.  Named directories may NOT be used
    in the definition.  Hence, a definition like $$ $0 A$
    A0 (for current disk/user, current disk/user 0, disk
    A/current user, and disk A/user 0) may be specified.

2.  Multiple Command Line Data

    a. Enable or Disable the Multiple Command Line Buffer
       If the Multiple Command Line facility is enabled by ZCPR2
       and the Cold Boot Routine in the BIOS, then the user
       should tell the utility that this facility is available.
       Note that the buffer MUST be initialized before use.
       Several utilities employ the Multiple Command Line Buffer
       for chaining purposes, and some utilities, like MENU and
       STARTUP, cannot run without it.

    b. If enabled, address of Multiple Command Line Buffer
       This specifies where the Multiple Command Line Buffer is
       located.  Note that only ONE Multiple Command Line Buffer
       may be defined for a particular system, unlike the
       external paths, in which case there may be as many as
       desired.  ZCPR2 is desired to deal with only one such
       buffer and cannot accept commands from more than one.
       Note, however, that many buffer areas may be defined for
       the purpose of passing information between utilities,
       particularly those who may be chained together via the
       Multiple Command Line Buffer Facility.  These utilities
       are chained via the buffer, but may use other buffers to
       pass information to each other.  They may also pass
       information to each other via the command lines in the
       Multiple Command Line Buffer.

3.  Maximum Disk which may be Referenced

    This parameter is available for the purposes of protecting the
    system and providing a level of security.  In the way of

protecting the system, by specifying the maximum disk letter
(like D if there are four drives), then the utility will flag
an error if a reference is made to any drive greater than D,
thereby trapping the error before the BDOS/BIOS sees it.

In the way of providing a level of security, certain programs
may be made available in two versions.  XDIR, for instance,
can be set up to be found along a user path in which it only
knows of drives A and B and does not permit access to C,
while, once a user has logged into a system directory and a
new, system path is established, a different XDIR may be
encountered along this system path which knows about drive C
also.

4.    Maximum User which may be Referenced

      This parameter is available for the purpose of providing a
      level of security only.   Along the lines of (3) above, two
      paths (user and system) and two sets of commands (like XDIR)
      may be established in which the anyone employing the user
      path cannot access beyond a given user number while anyone
      employing the system path can.

5.    Allow the User to Reference a Disk

      Again, to provide an even further level of security, the
      ability to reference any disk other than the current disk can
      be disabled.   This must be permitted in order for the D part
      of the DU form to be effective for the particular utility,
      and Named Directory references are affected as well.

6.    Allow the User to Reference a User Area

      This is the same as (5), but the ability to reference any
      user area other than the current user can be disabled.


7.    Privileged User Areas

      a. Base Address (first priv user area)
      A set of user areas, beginning at a particular user area, can
      be declared to be privileged.  Access to them is granted only
      if the correct password is given.  The recommended base
      address of the privileged user areas is 10, so user areas 10
      to 31 may be accessed only when a password is given.   Note
      that this option applies ONLY to the ZCPR2 utilities and not
      to ZCPR2 itself.   Once the ZCPR2 System is configured for
      this form of security, the DU form should be disabled within
      ZCPR2 or the Wheel facility should be set up to control use
      of the DU form.
      The privileged user definitions apply to the CD (Change
      Directory) program in particular.   This program is used in
      lieu of the DU: command to log into new directories.   It
      always checks to see if the user is entering a privileged
      directory and asks for the password before it allows him to
      enter.

      b. Password required to enter a priv user area

          This is the password required for access rights.

8.   Current User/Disk Symbol

     If the installer has opted to use some other symbol (other
     than "$") to represent the current disk or current user in
     symbolic path expressions under ZCPR2, the utility should be
     informed of this fact.   Naturally, some value has to be
     specified, and the recommended symbol of "$" is preferred for
     standardization purposes.

9.   DMA Address for Disk I/O

     If the user has a non-ORG-0 system, then this must be set to
     define the DMA Address (Default established by ZCPR2) of the
     system.  This address is usually RAM base + 80H, and the
     value for an ORG-0 system is 80H.


While running GENINS, which is menu-oriented, the user is provided
with the abilities to perform the following functions:

1.   Set any combination of the above characteristics for a
     particular utility

     The S option allows the user to sequentially pass through all
     of the necessary parameters and set them.  If the user wishes
     to select them individually, typing a digit from 0 to 9
     selects that particular parameter for installation.  If a
     particular option is not needed by the utility currently
     being installed, it is flagged with an asterisk, and
     selection of this option by digit performs no operation
     whatsoever.  The S option, if selected, simply skips all
     parameters flagged with an asterisk.
     Some utilities require parameters to be set which are not
     covered by the standard group.  Knowledge of the utility
     (refer to the User's Guide) is required to make a good
     judgment on the proper setting for the required parameter.
     If additional parameters are to be set, an A option appears
     in the menu.  The S option automatically extends to cover
     those utility-specific parameters, and the A option may be
     used like a digit option to select just the utility-specific
     parameters to be set.

2.   Display the current settings for a particular utility

     The P option displays the current settings of all parameters
     for the utility being installed.  If utility-specific
     parameters are available, these are printed as well.

3.   Save the current settings so that another utility may be set
     up in an identical way by means of a simple restore command

     The < option saves all of the standard parameters into to
     save buffer for later recall by the > option.  Using these
     options in the menu, the user can go through the task of
     setting all of the options of a "complete" utility (one in
     which all of the parameters indicated by digits are
     necessary) only once, then save his selections with the <
     option, write the current utility to disk, read in the next

utility without leaving GENINS, and then set its parameters
by the > option.  The next utility is then written to disk,
and a third utility is selected and its parameters set by >.
This can go on indefinitely, and this process significantly
reduces the time required to install the set of ZCPR2
utilities.

GENINS itself must be installed by GENINS, and, although
GENINS does not require all of the parameters, I have set it
up to declare all parameters necessary.  Hence, as a general
rule, install GENINS first and save your installation
settings via <.  Then, simply install each of the other
utilities in succession by using > to set the digit-specified
parameters and using the A option to set the utility-specific
parameters.


4.   Restore the settings from the last save as the current
     settings for a new utility

     From before, the > option is used to restore the parameters
     saved by the < option into a new utility.  If < has not been
     previously issued during the current GENINS run, then an
     error message will be printed if the user attempts to employ
     the > option.

5.   Quit and Do NOT Update the utility or Exit and Update the
     current utility

     If the user really blew the process of installing the current
     utility, he can simply throw away everything he did without
     affecting the utility file on disk.

6.   Set other attributes specific to the current utility which
     are not common to the other ZCPR2 utilities

     These parameters are noted when an A option appears in the
     menu.  These are unique to a specific utility or set of
     utilities and are not affected by the < and > commands (only
     the digit-selected parameters are affected by these
     commands).

6.1  Sample GENINS Session


The following GENINS session illustrates the process of utility
installation extensively.  Comments are noted out to the right side
as text preceeded by the "<<" symbol.



```
B4>genins genins          << Install GENINS first
GENINS Version 2.1        << This is probably NOT the version
                                << number which will be distributed
                                << with the ZCPR2 System
GENINS -- General Installation Program for ZCPR2 Utilities
Ready to Set Default Values   << It has found and read in the
                                << parameter section of the file


File Being Installed: GENINS  .COM
** GENINS Flag/Value Setup **      << Menu Follows
        Select Standard Flags and Values
                (* indicates unnecessary option) --


        P  Print Flags and Values      S  Setup ALL Flags and
Values
        <  Save All Flags/Values       >  Restore All Flags/Values
        Q  Quit and Do Not Update      X  Exit and Optionally
Update


        0  Setup Named Directory Data    5  Set Disk Allowed Flag
        1  Setup External Path Data      6  Set User Allowed Flag
        2  Setup Multiple Command Line   7  Setup Priv User Data
        3  Set Max Disk                  8  Set Current User/Disk
Data
        4  Set Max User                  9  Set File DMA Address


Setup Command (? for Menu)? P       << Print current params
**** Default Flag/Value Settings ****
Enable External Paths: Yes  External Path Address: 0040 Hex
   Internal Path is -- Empty
Enable Multiple Command Line Buffer: Yes  Buffer Address: FF00 Hex
Maximum Disk: D   Maximum User Number:  31
Disk Specification Allowed: Yes   User Spec Allowed: Yes
Base of Privileged User Areas:  10    Password: chdir
Current User/Disk Indicator: $   DMA Address for Disk I/O: 0080 Hex
Address of Memory-Resident Named Directory Buffer: FD00 Hex
    Number of Named Dirs Allowed:  64    Named Dir File: NAMES   .DIR


Setup Command (? for Menu)? S       << Set ALL Params in sequence


Address of Memory-Resident Named Directory Buffer: FD00 Hex
    Number of Named Dirs Allowed:  64    Named Dir File: NAMES   .DIR
** Setup Questions **
1  Address of Memory-Resident Named Directory Buffer
        Enter Hex Number or <RETURN> = None? fd00
2  Number of Named Directories Permitted?
        Enter Decimal Number or <RETURN> =  64?
3  Name of Named Directory File?
        Enter only File Name and Type -- DIR: form will be ignored
        Directory File Name (<RETURN> = NAMES    .DIR)?
```

```
Enable External Paths: Yes  External Path Address: 0040 Hex
    Internal Path is -- Empty
** Setup Questions **
1  Enable External Paths (Y/N/other=Y)?
1a Address of External Path in Hex (<RETURN> = 0040 Hex)?
2  Enable Internal Path (Y/N/other=Y)?
2a Define Internal Path Using Symbolic Notation --
1.  Path Element (<CR>=Done)? a$
2.  Path Element (<CR>=Done)? a15
3.  Path Element (<CR>=Done)?

Enable Multiple Command Line Buffer: Yes  Buffer Address: FF00 Hex
** Setup Questions **
1  Enable Multiple Command Line Buffer (Y/N/other=Y)?
1a Address of Multiple Command Line Buffer
        Enter Hex Number or <RETURN> = FF00 Hex?

Maximum Disk: D   Maximum User Number:  31
** Setup Questions **
1  Maximum Disk Allowed? B

Maximum User Number:  31
** Setup Questions **
1  Maximum User Number Allowed (<RETURN> =  31)? 15

Disk Specification Allowed: Yes   User Spec Allowed: Yes
** Setup Questions **
1  Specification of Disk Allowed (Y/N/other=Y)?


User Spec Allowed: Yes
** Setup Questions **
1  Specification of User Allowed (Y/N/other=Y)?

Base of Privileged User Areas:  10   Password: chdir
** Setup Questions **
1  Base of Privileged User Areas (<RETURN> =  10)?
2  Privileged User Password? chdir

Current User/Disk Indicator: $   DMA Address for Disk I/O: 0080 Hex
** Setup Questions **
1  Current User/Disk Indicator? $

DMA Address for Disk I/O: 0080 Hex
** Setup Questions **
1  DMA Address for Disk I/O
        Enter Hex Number or <RETURN> = 0080 Hex?


Setup Command (? for Menu)? P      << Print new settings
**** Default Flag/Value Settings ****
Enable External Paths: Yes  External Path Address: 0040 Hex
    Internal Path is -- A$: --> A 15:
Enable Multiple Command Line Buffer: Yes  Buffer Address: FF00 Hex
Maximum Disk: B   Maximum User Number:  15
Disk Specification Allowed: Yes   User Spec Allowed: Yes
Base of Privileged User Areas:  10   Password: chdir
Current User/Disk Indicator: $   DMA Address for Disk I/O: 0080 Hex
Address of Memory-Resident Named Directory Buffer: FD00 Hex
    Number of Named Dirs Allowed:  64   Named Dir File: NAMES   .DIR
```

```
Setup Command (? for Menu)? /       << Any invalid cmd gives menu
File Being Installed: GENINS  .COM << Reminder of file name
** GENINS Flag/Value Setup **
        Select Standard Flags and Values
                (* indicates unnecessary option) --


        P  Print Flags and Values      S  Setup ALL Flags and
Values
        <  Save All Flags/Values        >  Restore All Flags/Values
        Q  Quit and Do Not Update       X  Exit and Optionally
Update

        0  Setup Named Directory Data   5  Set Disk Allowed Flag
        1  Setup External Path Data     6  Set User Allowed Flag
        2  Setup Multiple Command Line  7  Setup Priv User Data
        3  Set Max Disk                 8  Set Current User/Disk
Data
        4  Set Max User                 9  Set File DMA Address


Setup Command (? for Menu)? <      << Save params



Setup Command (? for Menu)? X      << Exit and Update
Do you want to save changes (Y/N/other=Y)? Y      << Yes, Update
Please Provide Name of Output File
Name  of File (<CR>=GENINS  .COM)? genins2        << New  file
Name of Output File is GENINS2 .COM -- Verify (Y/N/other=Y)?
Writing File GENINS2 .COM to Disk --

Reading Block    95      << Numbers flash by as GENINS works
Writing Block    95
Please Provide Name of File to Install      << Ready to Continue
Name of File (<CR>=Abort)? mcopy       << New File
Ready to Set Default Values

File Being Installed: MCOPY   .COM
** GENINS Flag/Value Setup **
        Select Standard Flags and Values
                (* indicates unnecessary option) --


        P  Print Flags and Values      S  Setup ALL Flags and
Values
        <  Save All Flags/Values        >  Restore All Flags/Values
        Q  Quit and Do Not Update       X  Exit and Optionally
Update

        0  Setup Named Directory Data   5  Set Disk Allowed Flag
        1  Setup External Path Data     6  Set User Allowed Flag
        2  Setup Multiple Command Line  7  Setup Priv User Data
        3  Set Max Disk                 8  Set Current User/Disk
Data
        4  Set Max User                 9  Set File DMA Address

        A  Set MCOPY Special Default Values

Setup Command (? for Menu)? >               << Restore params from
                                            << previous setting
```

```
Setup Command (? for Menu)? P            << Print
**** Default Flag/Value Settings ****
Enable External Paths: Yes   External Path Address: 0040 Hex
    Internal Path is -- A$: --> A 15:
Enable Multiple Command Line Buffer: Yes   Buffer Address: FF00 Hex
Maximum Disk: B    Maximum User Number:  15
Disk Specification Allowed: Yes    User Spec Allowed: Yes
Base of Privileged User Areas:  10    Password: chdir
Current User/Disk Indicator: $   DMA Address for Disk I/O: 0080 Hex
Address of Memory-Resident Named Directory Buffer: FD00 Hex
    Number of Named Dirs Allowed:  64   Named Dir File: NAMES   .DIR
Verify by Default: Yes
Inspect by Default: No
Quiet by Default: No
Multiple Run by Default: No
Default Destination Disk/User: C  0:



Setup Command (? for Menu)? X        << OK, so exit and update
Do you want to save changes (Y/N/other=Y)? N       << No save
Please Provide Name of File to Install       << New file
Name of File (<CR>=Abort)? wheel         << File Name
Ready to Set Default Values

File Being Installed: WHEEL   .COM
** GENINS Flag/Value Setup **
        Select Standard Flags and Values
                (* indicates unnecessary option) --

        P  Print Flags and Values       S  Setup ALL Flags and
Values
        <  Save All Flags/Values        >  Restore All Flags/Values
        Q  Quit and Do Not Update       X  Exit and Optionally
Update

        0* Setup Named Directory Data    5* Set Disk Allowed Flag
        1* Setup External Path Data      6* Set User Allowed Flag
        2* Setup Multiple Command Line   7  Setup Priv User Data
        3* Set Max Disk                  8* Set Current User/Disk
Data
        4* Set Max User                  9* Set File DMA Address

        A  Set WHEEL Byte Address       << Note all the un-
                                        << necessary options and the
                                        << A option

Setup Command (? for Menu)? P
**** Default Flag/Value Settings ****
Base of Privileged User Areas:  10    Password: chdir
Address of Wheel Byte: 003B Hex

Setup Command (? for Menu)? X
Do you want to save changes (Y/N/other=Y)?
Please Provide Name of Output File
Name of File (<CR>=WHEEL   .COM)? wheel2
Name of Output File is WHEEL2  .COM -- Verify (Y/N/other=Y)?
Writing File WHEEL2  .COM to Disk --

Reading Block    10
Writing Block    10
```

```
Please Provide Name of File to Install
Name of File (<CR>=Abort)?



B4>diff  genins2.com,genins.com            << DIFF is good to
                                            << compare with
DIFF Version 1.6
Source File 1 -- B  4: GENINS2 .COM
Source File 2 -- B  4: GENINS  .COM
 Rel Offset   B  4: GENINS2 .COM     B  4: GENINS  .COM
  Hex    Dec      Hex  Dec Asc         Hex  Dec Asc
 0006     6        01    1  .           00    0  .
 0007     7        24   36  $           00    0  .
 0008     8        01    1  .           00    0  .
 0009     9        0F   15  .           00    0  .
 001A    26        02    2  .           04    4  .
 001B    27        0F   15  .           1F   31  .


B4>diff wheel2.com,wheel.com
DIFF  Version 1.6
Source File 1 -- B  4: WHEEL2  .COM
Source File 2 -- B  4: WHEEL   .COM
NO Differences Noted in Files
```

This Page Intentionally Left Blank

```
                            Appendix A

                     ZCPR2 Distribution Files
Disk 1 --
    1 File:  ZCPR2  .01  -- Size:      0K -- CRC: 00 00
        Comment:
    2 File: SYSIO   .ASM -- Size:     29K -- CRC: 91 11
        Comment: Sample Redirectable I/O Driver Package
    3 File: ZCPR2   .ASM -- Size:     56K -- CRC: 4F D1
        Comment: Source to ZCPR2
    4 File: ZCPRHDR .LIB -- Size:     15K -- CRC: E6 29
        Comment: Header/Configuration File for ZCPR2


Disk 2 --
    1 File:  ZCPR2  .02  -- Size:      0K -- CRC: 00 00
        Comment:
    2 File: DU2     .ASM -- Size:     72K -- CRC: 19 DA
        Comment: Disk Utility
    3 File: HELP2   .ASM -- Size:     33K -- CRC: 16 19
        Comment: Online Documentation System
    4 File: LRUNZ   .ASM -- Size:     26K -- CRC: B8 6F
        Comment: Library-Oriented Extended Command Processor
    5 File: SUB2    .ASM -- Size:     31K -- CRC: 57 FB
        Comment: Disk-Based Command File Processor/Ext Cmd Proc
    6 File: ZEX     .ASM -- Size:     45K -- CRC: 09 11
        Comment: Memory-Based Command File Processor/Ext Cmd Proc
    7 File: ZEX     .ZEX -- Size:      5K -- CRC: 53 28
        Comment: Command File Used to Assemble ZEX


Disk 3 --
    1 File:  ZCPR2  .03  -- Size:      0K -- CRC: 00 00
        Comment:
    2 File: CCPLOC  .MAC -- Size:      5K -- CRC: FA DA
        Comment: CCP/BDOS/BIOS Locator Program
    3 File: CD      .MAC -- Size:      9K -- CRC: F5 5D
        Comment: Change Directory (Log Into Named Directory) Util
    4 File: COMPARE .MAC -- Size:     13K -- CRC: 2F 8A
        Comment: File Compare Utility
    5 File: CRC     .MAC -- Size:     20K -- CRC: 4C 09
        Comment: CRC Utility
    6 File: DEVICE  .MAC -- Size:     13K -- CRC: 92 0A
        Comment: Redirectable I/O Driver Select/Display Utility
    7 File: DIFF    .MAC -- Size:     17K -- CRC: 74 C2
        Comment: File Difference Listing Utility
    8 File: ECHO    .MAC -- Size:      6K -- CRC: 99 64
        Comment: Command Line Echo Utility
    9 File: ERASE   .MAC -- Size:     15K -- CRC: 25 6B
        Comment: File Erase Utility
   10 File: GENINS  .MAC -- Size:     43K -- CRC: B2 18
        Comment: Utility Installation Program
   11 File: IOLOADER.MAC -- Size:      7K -- CRC: 4F 23
        Comment: Redirectable I/O Driver Package Loader
   12 File: LD      .MAC -- Size:     10K -- CRC: 6C D4
        Comment: Memory-Based Named Directory Loader
   13 File: LDIRZ   .MAC -- Size:     12K -- CRC: FF 1B
        Comment: Library Directory Utility
   14 File: MCHECK  .MAC -- Size:     14K -- CRC: B1 12
        Comment: Menu Syntax Check Utility
   15 File: MCOPY   .MAC -- Size:     37K -- CRC: BA 14
        Comment: Multiple File Copy with Named Directories Utility
```

```
Disk 4 --

    1 File:  ZCPR2   .04  -- Size:      0K -- CRC: 00 00
         Comment:
    2 File: MENU     .CPR -- Size:      1K -- CRC: 18 6F
         Comment: Sample MENU.CPR File
    3 File: MENU     .MAC -- Size:     24K -- CRC: 0D BE
         Comment: Menu Command Preprocessor
    4 File: MKDIR    .MAC -- Size:     21K -- CRC: BE 0A
         Comment: Named Directory Create/Edit Utility
    5 File: PATH     .MAC -- Size:     14K -- CRC: 9F 91
         Comment: Path Display/Set Utility
    6 File: PROTECT  .MAC -- Size:     18K -- CRC: EB 24
         Comment: File Protect Utility
    7 File: PWD      .MAC -- Size:     12K -- CRC: 1A 64
         Comment: Named Directory Environment Print Utility
    8 File: RECORD   .MAC -- Size:      7K -- CRC: DB 87
         Comment: Disk-Oriented Redirectable I/O Control Utility
    9 File: RENAME   .MAC -- Size:     18K -- CRC: 68 15
         Comment: File Rename Utility
   10 File: SINSFORM.MAC -- Size:      3K -- CRC: 3B E7
         Comment: General Header for ZCPR2 Utilities
   11 File: STARTUP .MAC -- Size:     10K -- CRC: D6 3F
         Comment: Command Line Insertion Utility
   12 File: TEMPLATE.MAC -- Size:     11K -- CRC: C5 DA
         Comment: ZCPR2 Utility Template
   13 File: TINIT    .MAC -- Size:     13K -- CRC: 9F 1E
         Comment: Terminal Init Program for TVI 950 CRT (uses Paths)
   14 File: WHEEL    .MAC -- Size:      6K -- CRC: 32 B6
         Comment: Wheel Set/Reset Utility
   15 File: XDIR3    .MAC -- Size:     39K -- CRC: 8C D9
         Comment: Extended Directory Display Utility
```

```
Disk 5 --
    1 File:   ZCPR2  .05  -- Size:      0K -- CRC: 00 00
    2 File: CCPLOC   .COM -- Size:      2K -- CRC: A0 7F
    3 File: CD       .COM -- Size:      4K -- CRC: CF FC
    4 File: COMPARE  .COM -- Size:      6K -- CRC: 00 1D
    5 File: CONFIG   .COM -- Size:     20K -- CRC: 44 4F
    6 File: CRC      .COM -- Size:      8K -- CRC: BB 23
    7 File: DEVICE   .COM -- Size:      4K -- CRC: 76 53
    8 File: DIFF     .COM -- Size:      6K -- CRC: E3 AF
    9 File: DU2      .COM -- Size:     10K -- CRC: 30 22
   10 File: ECHO     .COM -- Size:      2K -- CRC: C1 E9
   11 File: ERASE    .COM -- Size:      8K -- CRC: 00 7B
   12 File: GENINS   .COM -- Size:     14K -- CRC: 97 4D
   13 File: HELP     .COM -- Size:      4K -- CRC: 34 47
   14 File: IOLOADER.COM -- Size:       4K -- CRC: D3 C7
   15 File: LD       .COM -- Size:      6K -- CRC: F2 FC
   16 File: LDIRZ    .COM -- Size:      6K -- CRC: 98 7F
   17 File: LRUNZ    .COM -- Size:      2K -- CRC: B2 4D
   18 File: MCHECK   .COM -- Size:      6K -- CRC: A1 1D
   19 File: MCOPY    .COM -- Size:     12K -- CRC: 5A 81
   20 File: MENU     .COM -- Size:      2K -- CRC: DB E8
   21 File: MKDIR    .COM -- Size:      8K -- CRC: 22 60
   22 File: PATH     .COM -- Size:      6K -- CRC: ED 05
   23 File: PROTECT  .COM -- Size:      8K -- CRC: CB 21
   24 File: PWD      .COM -- Size:      6K -- CRC: B6 32
   25 File: RECORD   .COM -- Size:      2K -- CRC: 6E 42
   26 File: RENAME   .COM -- Size:      8K -- CRC: 5A 09
   27 File: STARTUP  .COM -- Size:      4K -- CRC: 98 04
   28 File: SUB      .COM -- Size:      4K -- CRC: 46 9D
   29 File: TEMPLATE.MAC -- Size:      12K -- CRC: C5 DA
   30 File: TINIT    .COM -- Size:      6K -- CRC: 60 62
   31 File: WHEEL    .COM -- Size:      2K -- CRC: D5 A0
   32 File: XDIR     .COM -- Size:     12K -- CRC: 1B E1
   33 File: ZEX      .COM -- Size:      6K -- CRC: 08 3E

Disk 6 --
    1 File:   ZCPR2  .06  -- Size:      0K -- CRC: 00 00
    2 File: CFILES   .HLP -- Size:     22K -- CRC: 10 97
    3 File: COMPARE  .HLP -- Size:      7K -- CRC: B3 F0
    4 File: CPMTR    .HLP -- Size:     24K -- CRC: 63 7B
    5 File: DU2      .HLP -- Size:     35K -- CRC: 5F 7B
    6 File: HELP     .HLP -- Size:      1K -- CRC: 15 97
    7 File: HELP2    .HLP -- Size:     22K -- CRC: 2D E1
    8 File: IO       .HLP -- Size:     14K -- CRC: D5 CA
    9 File: LU       .HLP -- Size:     27K -- CRC: 48 F2
   10 File: MCOPY    .HLP -- Size:     11K -- CRC: 96 9E
   11 File: MENU     .HLP -- Size:     29K -- CRC: 1C 39

Disk 7 --

    1 File:   ZCPR2  .07  -- Size:      0K -- CRC: 00 00
    2 File: NDIRS    .HLP -- Size:     35K -- CRC: F8 19
    3 File: XDIR3    .HLP -- Size:     22K -- CRC: 21 DA
    4 File: Z2ALT    .HLP -- Size:     22K -- CRC: 25 85
    5 File: Z2RES    .HLP -- Size:      7K -- CRC: FA D4
    6 File: ZCPR2    .HLP -- Size:      6K -- CRC: 73 08
```

This Page Intentionally Left Blank