WICAT Multi-user Control System

WMCS

System Manager's Reference Manual

188-190-201 C

May 1985

• Software •
Publications

WICATsystems

Receipt of this manual must not be construed as any kind of commitment, on the part of WICAT Systems Incorporated, regarding delivery or ownership of items manufactured by WICAT.

This manual is subject to change without notice.

first printing	August 1983
second printing	April 1984
third printing	May 1985

Typographical Conventions Used in this Publication

Bold facing indicates what you should type.

Square brackets, [], indicate a function key, the name of which appears in uppercase within the brackets. For example, [RETRN], [CTRL], etc.

Underlining is used for emphasis.

Information about this Manual

Review the following items before you read this publication.

The subject of this manual

The WMCS System Manager's Manual explains administrative procedures such as managing routine system operations, preventing loss of data, and handling problems.

The audience for whom this publication was written

The new or experienced system manager who has completed the tutorials in the WMCS System Manager's Introductory Manual.

Related publications

The chart on the following page lists other WMCS publications and the order in which they should be read.

Reader's Guide to WMCS Publications

Instructions: Determine the audience to which you belong and then read *only* the publications at an *arrowhead.* Dotted arrowheads indicate optional reading.

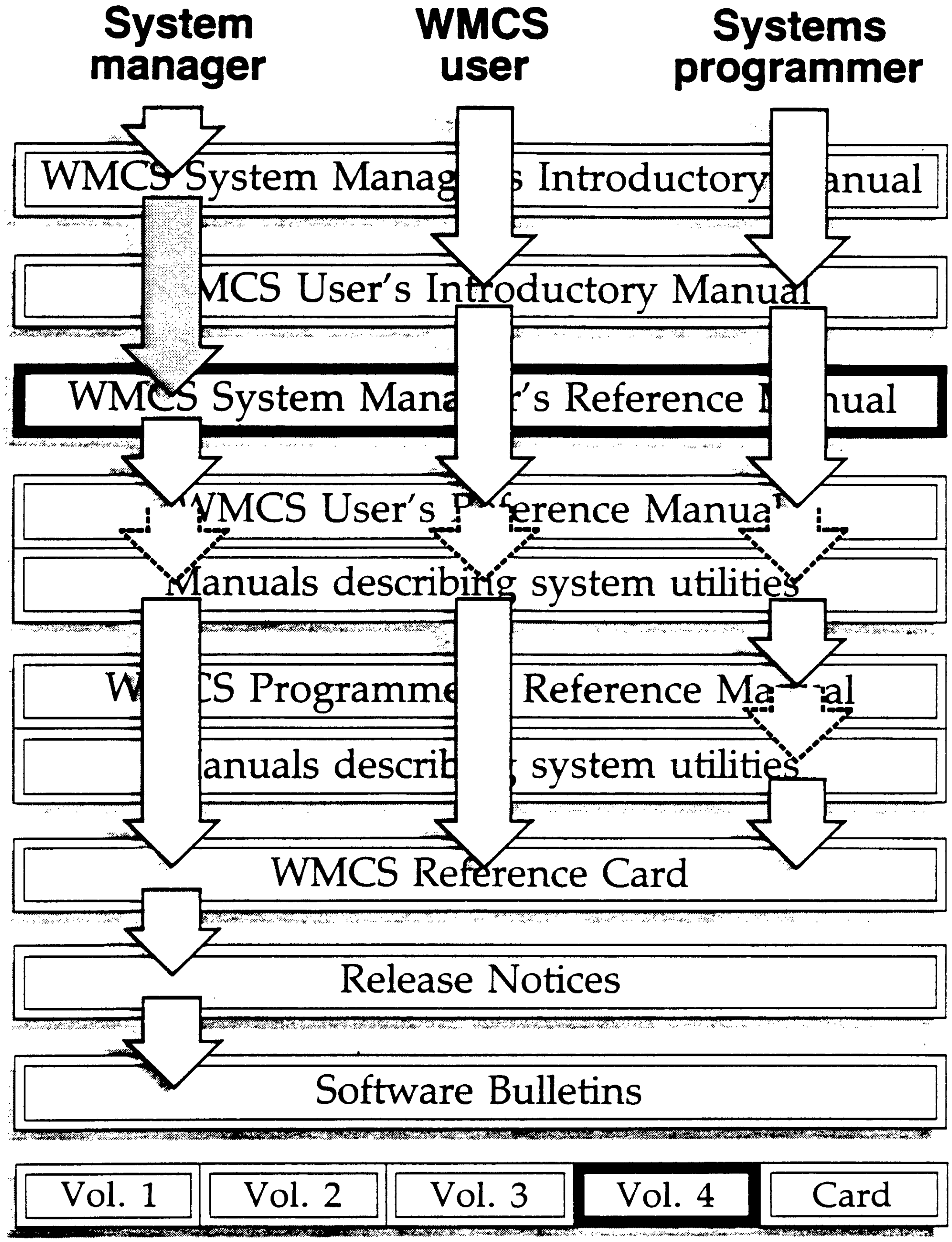

System manager
WMCS user
Systems programmer
WMCS System Manager's Introductory Manual
WMCS User's Introductory Manual
WMCS System Manager's Reference Manual
WMCS User's Reference Manual
Manuals describing system utilities
WMCS Programmer's Reference Manual
Manuals describing system utilities
WMCS Reference Card
Release Notices
Software Bulletins
Vol. 1
Vol. 2
Vol. 3
Vol. 4
Card

Table of Contents

Chapter 8 System Security and File Protection

Chapter 9 Initialization of Media

Chapter 10 Backup

Chapter 11 Turning Off the Power to the Computer

Chapter 12 How to Performance-Tune and Customize Your System

Chapter 13 Release Notices and Product Installation

Chapter 14 Configuring the WMCS

Chapter 15 Crash Recovery

Chapter 16 The System Disk

Chapter 17 The Memory Disk

Chapter 18 Networking

CHAPTER 1

THE DUTIES OF THE SYSTEM MANAGER

Inexperienced system managers should read this chapter.

One user on each WICAT computer should be designated the system manager. Where several computers are used at a single location, you may wish to designate a supervising system manager for the location and then assign a system manager to each computer at the site.

The system manager performs all administrative tasks associated with routine system operation. Nevertheless, the duties of the system manager do not constitute a full-time job. If possible, the system manager should have expertise in computer operations. However, the documentation set for the WICAT Multi-user Control System is designed so that even someone who is new to computers can serve as a system manager.

These are the responsibilities of the system manager:

1. Maintain the inventory of equipment pertaining to the system.

2. Formulate the schedule for system use, maintenance, etc.

3. Boot the system.

4. Perform daily, weekly, and monthly backups of software, and see that backup volumes are properly filed.

5. Set up user accounts, user-account default directories, and maintain system security.

6. Formulate command files, parameter files, logical name assignments, etc., to streamline and customize system use.

7. Ensure standardization of file designations, etc., among users.

8. Supervise the use of the system's disk space.

9. Ensure authorized use of the system.

10. Install all product releases.

11. Set the system's calendar clock.

12. Annually change the year in /ROOTDIR/SYSCONFIG.nnn.

13. Configure the WMCS.

14. Supervise the initialization of volumes, and the use of multiple drives on multi-user systems.

15. Maintain a log of system crashes, recurring diagnostic messages, hardware failure, etc.

16. Execute a recovery procedure following system crashes.

17. Arrange for hardware and software maintenance.

18. Be the only person authorized to turn off the power to the computer.

CHAPTER 2

EQUIPMENT INVENTORY

Find the Warranty Information form that was shipped with your WICAT Systems hardware.

You may wish to copy the completed form for each user on your system who will perform such tasks as initializing media, mounting serial ports, etc., so that those users will have the hardware information they need to execute various commands.

Update the form whenever a change in your system's hardware or software affects an item on the form.

CHAPTER 3

SCHEDULING USE OF THE SYSTEM

Both inexperienced and experienced system managers should read this chapter.

3.1 SCHEDULING SYSTEM USE

Inasmuch as each computer system supports only a certain number of users at a time, it is important to efficiently schedule system use. Consider this list as you develop a schedule to meet the needs of your system's users:

1. Determine when to leave the system on or off. Consider these points in making this determination:

   a. The period of time users need and actually use the system.

      For example, if the system will be idle during weekends or for any other extended period, you might consider turning the system off. Otherwise, leave the power on overnight to avoid the wear and tear that occurs from turning the power on and off.

      NOTE: If you decide to leave the system on overnight or for a long period during which the system will not be used, execute the SHUTDOWN Command to terminate all programs and dismount the disk drives. This provides some protection for the disk drives in case a power surge occurs.

   b. System protection.

Do not schedule use of the system when you, or another responsible and capable individual, are not present.

2. If you have more users than the system can accommodate at one time, consider the following items in formulating a schedule for shifts:

   a. The amount of time each day that each user needs access to the system.

   b. The system resources, i.e., memory, terminals, printers, disk space, etc., required by the software programs that will be used on the system.

   c. The importance of the jobs each user performs.

3. Always notify users of scheduled shutdowns, e.g., for maintenance, rebooting, etc., and (if possible) inform them of the duration of the interruption.

   The SHUTDOWN and SEND commands are described in the _WICAT Multi-user Control System (WMCS) User Reference Manual_.

4. Notify users when the system is rebooted and ready for use.

5. Put notices or messages to users in the /SYSLIB/LOGNEWS.TXT file. To have the message appear each time a user logs on, put a command like the following in the /SYSLIB/LOCALON.COM file:

   ```
   > type sys$disk/syslib/lognews.txt
   ```

   Whenever a user logs on to the system, LOCALON.COM is executed as part of that procedure and the messages in LOGNEWS.TXT appear on that user's terminal screen.

CHAPTER 4

BOOTING THE SYSTEM

Inexperienced system managers need to read only the following sections in this chapter:

When to boot the system

How to boot the system

Booting from a disk or diskette

4.1 WHAT IS THE SYSTEM BOOT?

When you turn on the power to your computer, or when you press the reset switch (read the operator's guide for your system), the microprocessor begins executing a boot program. The boot program, stored in your system's Read Only Memory (ROM), is the program whereby the machine is readied for use.

4.2 WHEN TO BOOT THE SYSTEM

Any one of the following situations requires that you boot the system:

1. The computer is off.

   The system must be booted whenever the power has been turned off. The WMCS is stored in main memory while the system is in use, and when power is turned off, the contents of main memory are lost.

2. You have reconfigured the WMCS, and want to use that reconfigured operating system.

The WMCS comprises modules such as the KERNEL, the KERNELBUG, the tape class handler, etc., that the boot program reads into main memory. If, once the WMCS has been read into main memory, you want to modify the group of modules that constitute the WMCS, you must reconfigure the WMCS (read the chapter in this manual on Configuring the WMCS) and then reboot the system.

Follow the instructions on the SHUTDOWN Command (in · the WICAT Multi-user Control System (WMCS) User Referercе Manual) **before** you reboot your system. Otherwise, you may lose data.

3. The system has crashed.

   You must reboot the system following a crash from which the WMCS does not recover (read the chapter on Crash Recovery to find out how to identify a system crash). Inasmuch as the power remains on unless the system crashed because of a power failure, you can use the reset switch to reboot following a crash.

   Follow the instructions in the chapter on Crash Recovery if the system crashes and you must reboot.

4.3 PRIMARY AND SECONDARY BOOT DEVICES

Systems shipped prior to May 1, 1984 contain two device drivers in the ROM, accessible to the boot program. The boot program can select one of the two devices to boot from by calling the driver that reads information from the selected device.

The first driver, designated the primary driver, is the default driver. It reads information from the primary boot device. The primary boot device is typically the system disk (a Winchester or SMD disk).

The second driver, designated the secondary driver, can be specified in place of the primary driver. The secondary driver reads information from the secondary boot device. The secondary boot device is always a removable media, such as a diskette or tape. See the Booting With Device Options section later in this chapter for a description of how to boot from the secondary device.

Having two drivers to choose from gives you considerable flexibility in booting your system. For example, suppose that version 5.0 of the WMCS is the operating system that is initialized when you boot from your primary device. However, you want to boot with version 4.2.0 of the WMCS, a version you have on a set of diskettes. By following the appropriate procedure, you can use your floppy-disk drive to boot from the floppies.

Being able to boot from the secondary device becomes extremely important when you are unable to boot from the primary device. Booting from the secondary device then allows you to mount the primary device, correct the problem, and reboot the system from the primary device.

Three situations may prevent the system from being booted by the primary device:

1. A file on the hard disk that is needed to boot the system may be damaged. Error messages should provide a clue as to which file may be damaged.

   To overcome this problem:

   a. Boot from the secondary device

   b. Mount the hard disk.

   c. Correct or replace the bad file.

2. The file system is corrupt.

   To overcome this problem:

   a. Boot from the secondary device.

   b. Mount the hard disk.

   c. Execute the RECOVER Command to rebuild the file system. See the WICAT Multi-user Control System (WMCS) User Reference Manual for instructions on how to use RECOVER.

3. You tried booting from the secondary device and still cannot mount the hard disk.

   To overcome this problem:

   a. Re-initialize the disk. See the WICAT Multi-user Control System (WMCS) User Reference Manual for instructions on how to use the DINIT Command.

   b. Use the RESTORE Command to replace the files from your backups.

The boot process is the same whether you boot from the primary or the secondary device.

Systems shipped after May 1, 1984 use the universal boot. These systems are no longer restricted to two device drivers.

The universal boot ROMs contain drivers for every possible WICAT-supported boot device. Since the ROMs are independent of drive type, there need only be one set of ROMs for each product family. Therefore, the universal boot ROMs for any system 150, 155, or 160 are identical even though one boots from a 10 megabyte Winchester and the other boots from a 474 megabyte SMD.

The System 140 has two possible boot devices—5.25-inch floppy disk and Winchester disk.

The System 150 has five possible boot devices—5.25-inch floppy disk, 5.25-inch Winchester disk, SMD disk, cartridge tape, and Cipher tape.

The System 200 has six possible boot devices—5.25-inch floppy disk, 5.25-inch Winchester disk, SMD disk, cartridge tape, Cipher tape, and 8-inch floppy disk.

4.4 HOW TO BOOT THE SYSTEM

4.4.1 Booting With The Default Device

1. Use one of the following methods to begin the boot process:

   a. If the machine is off, turn the Power Control to the ON position.

   b. If the machine is on and already booted, and you need to reboot the system, use the SHUTDOWN Command and specify the :REBOOT Switch.

   c. If the system has crashed, press the Reset Control.

2. The following message appears in the upper left-hand corner of the terminal screen (for the terminal attached to serial port _TT0) when the boot process has begun:

   ```
   Booting...
   ```

   Other log messages then appear as the system goes through the boot process described above. The boot process has been completed when the following report appears on the screen:

```
(boot$cip) SYSTEM NAME is up and running.
```

The system is ready for users to log on.

4.4.2 Booting With Device Options

Follow this procedure to boot from your system's secondary device, i.e., a diskette or tape:

1. Use one of the following methods to begin the boot process:

   a. If the machine is off, turn the Power Control to the ON position.

   b. If the machine is on and already booted, and you need to reboot the system, use the SHUTDOWN Command and specify the :REBOOT Switch.

   c. If the system has crashed, press the Reset Control.

2. Press the spacebar when the following message appears on the screen (for the terminal connected to serial port _TT0):

   ```
   Booting...
   ```

   After a moment, one of the following prompts appears:

```
Booting secondary device.  Insert diskette and press [RETRN].
```

or

```
Boot failed.  Enter boot id (P/S, drive id):
```

3. If the first prompt appears, insert the diskette (or tape) in the correct drive. Then skip steps 4 through 6.

4. If the second prompt appears, make sure your secondary boot volume, i.e., diskette or tape, is loaded in the correct drive.

5. Refer to the system inventory form you completed when the system arrived and determine the secondary device driver identification number.

6. In response to the "boot id" prompt, type s followed by the device drive number as shown in the following example:

```
Boot failed.  Enter boot id (P/S, drive id):  s0b0
```

7. Strike [RETRN].

   A series of boot log messages appears on the screen.

   When the boot procedure is finished, you will not need to log on. The cursor appears next to the right angle bracket.

Follow this procedure for the universal boot:

1. Use one of the following methods to begin the boot process:

   a. If the machine is off, turn the Power Control to the ON position.

   b. If the machine is on and already booted, and you need to reboot the system, use the SHUTDOWN Command and specify the :REBOOT Switch.

   c. If the system has crashed, press the Reset Control.

2. Press the spacebar when the following message appears on the screen (for the terminal connected to serial port _TT0):

```
     Booting...
```

   After a moment, a prompt like the following appears:

```
Available drive types are:
     F5 -- 5 1/4 inch Floppy disk
     W5 -- 5 1/4 inch Winchester disk
Selection:
```

   NOTE: The list of available drive types will vary depending on the types of controller boards in your system.

3. Type the two-digit code (next to the "Selection:" prompt) that applies to your drive type and strike [RETRN]. A prompt like this appears:

   ```
   Drive #, Board # (d,b):
   ```

4. Type the drive number, a comma, and the board number. Then strike [RETRN].

   The drive number indicates which of possibly several drives you wish to boot from. For instance, your system may have two Winchester disks, drive 0 and drive 1. Type either a 0 or a 1 for the drive number.

   The board number indicates the controller board for the drive. Some device drivers support multiple controller boards. For instance, your system may have two SMD controller boards, 0 and 1. Controller board 0 supports up to four disks (0 through 3 on board 0). Controller board 1 also supports up to four drives (0 through 3 on board 1).

   If you want to boot from drive 1 on controller board 0, type **1,0** and strike [RETRN].

   The default for the above prompt is 0,0 and will be assumed if you strike [RETRN] without typing any numbers. A series of boot log messages will then appear on your screen.

4.5 FILES THAT MUST BE ON THE BOOT VOLUME

The boot volume may be a disk, a diskette, or a tape. Certain files must be on each boot volume. With a boot tape, these files must be on the tape in the correct order. See the appendixes in the back of this manual for sample command files that can be used to build a boot volume.

4.5.1 Disk Or Diskette

The volume you use to boot your system must contain these files:

/ROOTDIR/	/SYSEXE/	/SYSLIB/	/SYSDSR/
KERNEL.nnn	CIP.EXE	STARTUP.COM	Device driver for boot device
SYSCONFIG.nnn			TTY$nnn.DSR
DISK.nnn			NULL.DSR
TTY.nnn			
OSINIT.nnn			
STARTUP.nnn			
BOOTDISK.nnn			
DEVCONFIG.nnn			

Note that the foregoing files allow you to boot, i.e., if the foregoing files are the only files on the disk, all you can do is boot; you cannot execute other commands because no other commands are on the disk.

Furthermore, SYSCONFIG.nnn must refer to only those files that are on the disk.

Finally, STARTUP.COM must fork CIP (&CIP).

Appendix C contains a sample command file that can be used to build a boot diskette.

4.5.2 Tape

These files must be present on the volume in the following order:

```
/ROOTDIR/BOOTTAPE.nnn
/ROOTDIR/TAPCONFIG.nnn
/ROOTDIR/KERNEL.nnn or /ROOTDIR/KERNELBUG.nnn
/ROOTDIR/DISK.nnn
/ROOTDIR/TAPE.nnn
/ROOTDIR/TTY.nnn
/ROOTDIR/OSINIT.nnn
/SYSDSR/<tapedriver>.DSR
/SYSDSR/TTY$nnn.DSR
/SYSDSR/NULL.DSR
/ROOTDIR/STARTUP.nnn
/SYSEXE/CIP.EXE
```

Appendix D contains a sample command file that can be used to build a boot tape.

4.6 BOOTING FROM A DISK OR DISKETTE

1. The microprocessor executes the boot program stored in your system's ROM. This program does three things:

   a. Performs a cursory analysis of the system, e.g., prepares your system's memory for use; determines the number of available Universal Asynchronous Receiver Transmitters (UARTs, or TTY ports); checks the system clock, the calendar clock, the memory mapping registers, etc. (This cursory check is performed only on initial power up.)

   b. Links all pages of memory into a usable list of pages, and gives each TTY port a standard format.

   c. Reads the BOOTDISK.nnn file from the boot disk (nnn is a variable corresponding to the hardware system you have, e.g., 156 for a System 150-6WS, etc.), writes that file to memory, and transfers control of the system to that file.

      These are the steps the boot program performs to read the BOOTDISK file:

      (1) Reads the boot block from the disk. There is one boot block on each disk, at sector 0. The boot block contains the sector number assigned to the first sector of FCB.SYS.

      (2) Finds and reads the FCB for the ROOTDIR.DIR file. This FCB contains information on the physical location of the ROOTDIR.DIR file.

      (3) Scans the ROOTDIR.DIR file to find the record for the highest numbered version of the BOOTDISK file. That record contains the FCB number for the BOOTDISK file.

      (4) Goes to FCB.SYS and reads the FCB for the BOOTDISK file. That FCB contains information on the file's physical location on the disk.

      (5) Reads the BOOTDISK file from the disk and write it to memory.

2. BOOTDISK reads the SYSCONFIG.nnn file to find out what files constitute the WMCS, finds the files on the disk, reads them from the volume, writes them to

memory, and transfers control of the system to those files. It also builds a table of all class handlers and loads the boot device driver (read the chapter on Configuring the WMCS for details on the format of the SYSCONFIG.nnn file).

OSINIT.nnn, which is described next, is one of the files written to memory.

NOTE: All diagnostics reported up to this point in the boot process do not correspond to WMCS diagnostic numbers. The diagnostic numbers used during the boot process are defined in Appendix B.

3. OSINIT performs several functions:

   a. Initializes memory.

   b. Sets the year and loads the tick clock.

   c. Computes the size of the operating system.

   d. Sets up the Process Control Block (PCB) for the initial process.

   e. Turns on clock interrupts.

   f. Sets up the initial process and transfers control to the WMCS, which begins scheduling processes.

4. The initial process set up by OSINIT completes the following functions:

   a. Mounts the boot device (the drive in which the boot volume is located).

   b. Mounts the lowest numbered good TTY port; usually _TT0.

   c. Assigns logical names to SYS$DISK, SYS$MODEL, SYS$PROMPT, and SYS$MATH.

   d. Assigns the default device and directory to the initial process.

   e. Mounts the _NULL device.

The initial process then dies after forking the execution of this file:

```
/ROOTDIR/STARTUP.nnn
```

It is strongly recommended that you use the standard system startup file that loads and runs CIP, instead of running a turnkey application.

5. STARTUP also dies after it forks a noninteractive version of the CIP that executes this command file:

   ```
   /SYSLIB/STARTUP.COM
   ```

   Do not modify the STARTUP.COM file. The LOCALUP.COM file should be used to customize your system.

4.6.1 Booting From A Tape

1. The microprocessor executes the boot program stored in your system's Read Only Memory (ROM). This program does three things:

   a. Performs a cursory analysis of the system, e.g., prepares your system's memory for use; determines the number of available Universal Asynchronous Receiver Transmitters (UARTs, or TTY ports); checks the system clock, the calendar clock, the memory mapping registers, etc. (This cursory check is only performed on initial power up.)

   b. Links all pages of memory into a usable list of pages, and gives each TTY port a standard format.

   c. Reads the BOOTTAPE.nnn file from the boot tape (nnn is a variable corresponding to the hardware system you have, e.g., 156 for a System 150-6WS, etc.), writes that file to memory, and transfers control of the system to that file.

   The boot program reads the first file on the tape whether or not its name is BOOTTAPE.nnn. The BOOTTAPE file is responsible for loading the rest of the WMCS into memory, and requires that the WMCS files be on the tape in a predefined order.

2. BOOTTAPE reads the TAPCONFIG.nnn file, which must be

the second file on the tape, to find out how many files to read from the tape after reading the KERNEL file and before reading the OSINIT file. It then reads the specified number of files from the tape, writes them into memory, and transfers control of the system to those files.

The BOOTTAPE file assumes the following sequence of files (where the files indicated with an asterisk, *, are specified by the TAPCONFIG file):

```
/ROOTDIR/KERNEL.nnn   or   /ROOTDIR/KERNELBUG.nnn

* /ROOTDIR/DISK.nnn
* /ROOTDIR/TAPE.nnn
* /ROOTDIR/TTY.nnn
* /ROOTDIR/KSAM.nnn

/ROOTDIR/OSINIT.nnn
```

Even though the boot tape program pays no attention to the names of the files, you should use the standard file designations indicated above.

Three device drivers that are required to bring up the system must appear on the tape immediately after OSINIT. These three files are the tape device driver, the terminal device driver, and the null device driver. The name of the tape device driver is not checked, but the other two files must have the correct names:

```
/SYSDSR/<tapedriver>.DSR
/SYSDSR/TTY$nnn.DSR
/SYSDSR/NULL.DSR
```

Immediately following the device drivers, the following sequence of files must appear on the tape so that the WMCS will be able to complete the boot process:

```
/ROOTDIR/STARTUP.nnn
/SYSEXE/CIP.EXE
```

OSINIT.nnn is one of the files written to memory.

3. OSINIT performs several functions:

    a. Initializes memory.

    b. Sets the year and loads the tick clock.

c. Computes the size of the operating system.

d. Sets up the PCB for the initial process.

e. Turns on clock interrupts.

f. Sets up the initial process and transfers control to the WMCS, which begins scheduling processes.

4. The initial process built by CSINIT completes the following functions:

   a. Mounts the boot device (the drive in which the boot volume is located).

   b. Mounts the lowest numbered good TTY port; usually _TT0.

   c. Assigns logical names to SYS$DISK, SYS$MODEL, SYS$PROMPT, and SYS$MATH.

   d. Assigns the default device and directory to the initial process.

   e. Mounts the _NULL device.

   The initial process then dies after forking the execution of this file:

   /ROOTDIR/STARTUP.nnn

   It is strongly recommended that you use the standard system startup file that loads and runs CIP, instead of running a turnkey application.

5. STARTUP also dies after it forks an interactive version of the CIP.

CHAPTER 5

SYSTEM TIME

5.1 THE CALENDAR CLOCK

Use the TIME Command to set the time on the system's calendar clock. This command is described in the WICAT Multi-user Control System (WMCS) User Reference Manual.

Inasmuch as all timed aspects of system operation, such as the creation and modification dates of files, etc., is based upon the calendar clock, only the system manager should be permitted to set the time. For example, the backup of files will not work correctly if system time is incorrect.

If your system does **not** have a calendar clock, the following command line character string should appear in SYS$DISK/SYSLIB/LOCALUP.COM:

```
> TIME :PROMPT
```

5.2 ANNUALLY CHANGE THE YEAR IN THE SYSCONFIG.NNN FILE

The line in SYSCONFIG.nnn (nnn varies according to the system you have, e.g., SYSCONFIG.156 for a System 150-6WS, etc.) immediately before the line containing the file designation for the device driver for the primary device, contains the year used by the calendar clock. The clock keeps track only of day, month, hours, minutes, etc., not the year. Therefore, on the first working day of the year you should use the SYSPROF Command to access SYSCONFIG.nnn and change the year in that file.

Chapter 6

System Security and User Accounts, Part

This chapter is for inexperienced and experienced system managers.

What is System Security?

The WMCS provides several ways to maintain the security of your system. System security includes:

1. Limiting access to the system to authorized users.

2. Controlling what authorized users can do once they gain access to the system, such as which files and devices they can modify and which processes they can affect.

How to Enhance System Security

Take the rollowing measures regarding system security:

1. Assign a password to the system manager's account (follow the procedure given in the WMCS Introductory System Manager's Manual).

2. Assign a password to every account you create.

   Passwords should be at least eight characters long. This reduces the probability that someone will, through a process of elimination, happen upon an authorized username and password. Therefore, instruct the users on your system to specify long passwords whenever they reassign a password to their accounts.

3. Make sure your system is physically secure, e.g., accessible only to users.

4. Have a modem on your system only when you can monitor its use.

5. Use the directory hierarchy (when necessary) to make it difficult to identify proprietary material.

What is a User Account?

A user account is a record that the WMCS maintains for each username that you want the WMCS to recognize.

When your system arrives, UAF.DAT contains the following records:

1. DEFAULT

2. SYSTEM

The DEFAULT record is like a blank form at the front of a file drawer that you use each time you add a user account to the file; the DEFAULT record is a template. The SYSTEM record is the user account for the system manager.

Each time you create a user account, you add a record to the UAF.DAT file.

When a person types a string of characters in response to the prompt for a username, the WMCS searches UAF.DAT to find a "card" or record on which that character string appears as the username. The WMCS then uses the information on that "card" or record to create a user process.

See the description of the USERPROF Command in the WMCS User's Reference Manual for information on how to create and modify user accounts.

Deletion of User Accounts

Use the USERPROF Command to delete the User Authorization Record (UAR) for the expired account.

Delete (or otherwise dispose of) the files in the user-account default directory associated with the old account.

You may want to preserve some or all of the files associated with the deleted account. You can move them to another working directory with the REN Command, or you can back them up on a diskette or tape.

See the WMCS Introductory System Manager's Manual for a tutorial on creating user accounts and user-account default directories.

Chapter 7

System Security and User Accounts, Part 2

The material in this chapter is for inexperienced and experienced system managers.

Why is System Security Important?

There are many reasons to be concerned with system security. Your system may contain confidential data (financial data, personnel records, government records, licensed software products, etc.). Or, your system may have data files or programs that are critical to the operation of your system. If inexperienced or malicious users accidentally modified some important tile, they could cause the system to malfunction.

One of your responsibilities as a system manager is to protect the data on your system trom unauthorized inspection, theft, or tampering.

This chapter and the next define the features available in the WMCS that will help you with this responsibility. This chapter deals with three major areas:

1. Controlling access to the system using LOGON

2. Defining ownership of resources (The User Identification Code)

3. Process privileges

A good understanding of these principles is essential to system protection.

It is important to realize that there are very few, if any, foolproof mechanisms for protecting data. It is very much like keeping your car from being stolen. You can discourage the potential burglar by locking the car and taking the keys. But a determined, informed burglar can still find a way to steal it, if he tries hard enough.

Your challenge is to take precautions to make unauthorized tampering difficult while not restricting legitimate system usage.

System security is concerned with two basic ideas:

1. Controlling access to the machine. This includes not only physical access (i.e., who can get into the same room with the machine) but also log on access (i.e., who can log on to the system to access its files).

2. Controlling what users can do once they have gained access to the machine. For instance, a student may be an authorized system user, and he may have access to various lessons, or compilers. He should not have access to the files containing grades or test questions.

Controlling Physical Access to the System

Limiting physical access to the machine involves common sense and has nothing to do with the WMCS. Here are two suggestions:

1. Secure the facilities that contain the computer. Keep the computer and any terminals connected to the computer locked in an office when they are not being used.

2. Modems provide the potential for accessing the system across phone lines, even though it is locked away. You might consider connecting a modem to your system only during times when it is being used for legitimate business, and when you can monitor its use. (Other issues related to modems are discussed later.)

Controlling Logon Access to the System

The basic LOGON process in the WMCS is simple and straight forward. LOGON displays a prompt for a username, and optionally a password. The prospective user identifies himself by responding to those prompts. LOGON checks that the user's response matches one of the user authorization records in the user authorization file. If it does, then LOGON grants access to the system by creating the process specified in that user authorization record. Usually this process is the CIP, but you will see later how to change this.

The Role of Passwords

The password plays a key role in qualifying the prospective user. For proper security, passwords must be kept secret. They are not displayed on the screen as the user types them. They are recorded in the user authorization file in an encrypted form. Since they are encrypted, there is no easy way to find out what the password is for any given account.

There are two ways of assigning a password to an account. The first way involves using the USERPROF utility as described in the previous chapter. The second way is using the PASSWORD utility. Once a user has successfully logged on, he can use the PASSWORD utility (described in the User Reference Manual) to modify the password for his account. This allows users to change their password as often as they like without having to burden the system manager to change it with USERPROF.

You may want to record the password of the system manager's account. You should not need to keep a record of any other passwords, because if you can log on to the system account, you can set the password to any other account. If you do record the system password, keep it in a safe place, away from the computer system. Do not store it in a file on the system disk.

Password selection can help control access. Here is your opportunity to be creative. Select passwords that would be difficult to guess. The best choices are words that do not appear in the dictionary and are more than four characters long. Your initials, your spouse's name, and your last name are poor choices for passwords.

Safeguards Built into LOGON

There are several safeguards built into LOGON to help prevent password stealing and log on by exhaustive search.

One classical method of password stealing involves writing a program that mimics LOGON. This program would look for all the world like the real LOGON program. The unsuspecting user types his username and password. The program then records the password in a file, and disappears. The user thinks that since he was not logged on, he must have made a typographical error. So he tries to log on again. This time the real LOGON program comes up, and the user logs on successfully, never suspecting that his password may have been stolen.

Log on by exhaustive search involves attaching another computer to a terminal port and having one computer try to log on to the other by trying all combinations of possible usernames and passwords.

LOGON does the following things to minimize the threat of these two problems:

1. A welcome banner is displayed as LOGON begins. The purpose of the welcome banner is not just to make the user feel more welcome, but also to notify the user that LOGON has just begun. If the user tries once to LOGON, and fails, and then he tries again, and the welcome banner appears again, the user then has some cause for suspicion. His password may have been stolen. He can then easily log on and change his password.

2. The user is given five attempts to correctly type his username and password. The welcome banner comes up only once, just before the first attempt. If the password stealing program tries to mimic this, the user also becomes suspicious. He may have made a typographical error once, or maybe twice. But the third time, he was very careful. If he tries five times and fails, and then tries again and makes it, he might suspect that his password has been stolen.

3. If the user specifies a username for which there is no account, LOGON automatically prompts for a password anyway. This is an attempt to not give any clues to computers doing an exhaustive search. They can't tell if the username they specified was valid or not.

4. After five unsuccessful attempts, LOGON is disabled on that TTY port for 30 seconds. The idea behind this is that a human user will usually get his password right within five tries. A computer, trying an exhaustive search, will have to try hundreds or thousands of combinations before it finds one that works. By giving it only five tries every 30 seconds we have significantly increased the amount of time it takes to try those combinations.

Options Available With LOGON

Using the SYSPROF command (See the WMCS User's Reference Manual for details) the system manager has several options available. SYSPROF allows you to modify the device configuration file (DEVCONFIG.nnn). The device configuration file contains one record for each device on your system. There are three fields that are used with LOGON ports that specify which users can log on at that port.

1. :defaultuser=

   This field is used on a terminal port where you do not want to have to log on. If the :defaultuser= field is specified, a person can log on at that terminal port by merely pressing the RETURN key.

2. :authorized=

This field specifies a list of usernames that will be allowed to log on at this port. This switch might be useful on a modem port. It can be used to restrict log ons on that port to a subset of the total community of authorized users.

3. :exclude=

   This field specifies a list of usernames that are not allowed to log on at this port.

Suppose that _tt3 is in your office. When you are not there, your office is locked. You are the only person that uses that line. You can set up the :defaultuser= field in the record for _tt3 of the device configuration file to be your username. Then, when you want to use the system, you can go to _tt3 and log on by merely pressing the return key. If your username is JONES, then you will be logged on to _tt3 as JONES.

NOTE: Using the :defaultuser= switch creates a big hole in the system security. In the above example, anyone who can connect a terminal to _tt3 (whether it is the terminal in your office or not) can log on to the system without having to specify a password. This mechanism is, however, very useful for non-security sensitive systems.

Suppose that _tt2 is connected to a phone line through a modem. George and William, who work in another building, occasionally need to use your system. You would like for them to be able to call into your system through the modem. To minimize the risk of having a modem attached to your system, you use SYSPROF to assign the value GEORGE,WILLIAM (assuming that those are their usernames) to the :authorized= switch of the record for _tt2 of the device configuration file.

The Security Program Option

There are some installations that would like to have special-purpose user verification programs. The password facility is insufficient for their needs. Using the USERPROF program, the system manager can specify a user verification program that replaces the password facility. For instance, some military installations have a password code that involves a more dynamic interaction than just typing a password.

When LOGON executes, it asks the user to type in his username. Then it locates the record in the user authorization file that corresponds to that username. If the SECURITY field of that record is not empty, then instead of prompting for a password, LOGON creates a process (the filename of the image file is specified in the security field). That process can perform any type of verification check. When the verification program terminates, it calls the _exproc system call. (Note that the result parameter of _exproc is returned to the parent process in the ccode parameter of the _crproc system call. See the WMCS Programmer's Reference Manual for details.) If the result parameter of the _exproc system call is zero, LOGON assumes that the user passed the verification test, and proceeds with the log on. If the result parameter is non-zero, LOGON assumes failure.

The security program can be as complicated or as simple as you want. It can use special input devices (voice recognition systems, finger print analysis systems, etc.).

Suppose that at your facility you would like to have an interactive user verification program. You make up a list of codes. The system will give the user a code word, and the user must respond with the corresponding code. For instance, the code table might be as follows:

Computer code	Correct response
blue	banana
orange	grape
red	peach
green	cherry
violet	melon
yellow	strawberry
black	apple

To implement this, you write a program that will randomly select one of the computer codes and then ask the user to supply the correct response. If the user's response matches the expected response, then your program terminates with a result of zero. Otherwise it returns with a non-zero result. The following is a typical example:

```
Welcome to Wally
Username: james
Yellow  : strawberry
```

"Wally" is the system name. The user specifies "james" as the username. LOGON checks the account for james and sees that a security program was specified, so it initiates that program. The security program selects "yellow" as the code of the day. The user correctly replies with "strawberry." The security program terminates with a result of zero, and the user is logged on.

Ask System Users to Log Off

To provide a reasonable amount of system security you should request that all ot your system users log off whenever they leave their terminals unattended. This prevents unauthorized users from using a terminal that is already logged on. Or, you can start up the WATCHDOG program on your system. This program automatically logs off terminals that have been idle. See the WMCS User's Reference Manual for a description of WATCHDOG.

Restricted Access Accounts

A restricted access account is a LOGON account that does not produce a CIP. CIP, of course, is a general purpose command interpreter. As such, any user that logs on to CIP can use many of the utility programs to rummage through the file system (as far as permitted by file protection and process privileges discussed below).

There may be some users of your system whose system requirements are restricted to a word processor or some other application. For such users you can set up an account that brings up the application instead of the CIP. When the user exits from the application, they are logged off. In other words, the user's access to the system is restricted to the things they can do from within that application.

You can also set up a restricted access account that executes a CIP command file, and then logs off when the command file finishes. The principle is the same. The user's access to the system is restricted to the things that are done in that command file.

Why would you want to set up a restricted access account? Suppose that your system users need to create tapes using the COPY utility program. The COPY program requires that the tape be mounted :special. To mount a tape with the :special option requires operator privilege (see discussion of privileges below). You would rather not give all of your system users operator privilege, so you create a restricted access account that has operator privilege. When ever a user logs on to this account, a command file is executed which will mount the tape and perform the COPY according to the user's needs. When the command file finishes, the user is logged out of that privileged account. In other words, users can log on to this privileged account, have the privileges they need for a restricted operation, but not be able to use that privilege for any other operations.

Here are two examples of how to set up a restricted access account:

1. Suppose that the restricted access account executes a specific application. Use the USERPROF command, described in the previous chapter, to add a new account. Then on the line labeled "Command line" type the command line that will bring up that application. For instance, if the application were the Q-One word processor, then the command line might read "cip @sys$disk/usr.qlib/qone;log :nolog". When a user logs on to this account, he comes up in Q-One instead of CIP. When he exits Q-One, he will be logged off the system.

2. Suppose that the restricted access account executes a command file. Use the USERPROF command, described in the <u>WMCS User's Reference Manual</u>, to add a new account. Then, on the line labeled "Command line" type a command like the following:

```
cip @sys$disk/syslib.users/comfile.com; log :nolog
```

This command line will initiate a CIP to execute the command file named "comfile.com". When the command file finishes, (or is aborted with a [CTRL] c) then the log command is executed to terminate the CIP that was created. The :nolog switch is for cosmetics only. It suppresses the log message that normally appears on the terminal when a user logs off.

Controlling the Use of Modems

As mentioned above, modems connected to your system pose a threat to security.

It doesn't matter how securely you have locked away your computer. Both authorized and unauthorized users can attempt to log on through the modem.

An unauthorized user can call into your system with another computer and attempt to log on to your system using an exhaustive check of all usernames and passwords.

An authorized user may have used the modem, logged on, and then disconnected without logging off. An unauthorized user may then call into the machine through the modem, and have access to the machine without logging on.

There are several things that you can do to minimize the threat posed by modems:

1. Use the :authorized field in the DEVCONFIG file (as described above) to specify a limited set of accounts that can be used on the modem port.

2. Use a more sophisticated security check program (as described earlier) on all accounts that can be used on the modem port.

3. Some of the I/O ports on WICAT equipment allow modem control. With these ports you can use the DSTAT command to set the modem port to :REMOTE and :MODEMCTRL. With these two options set, the WMCS will automatically terminate all processes associated with that port when the telephone line is disconnected. This prevents unauthorized users from calling in and using the machine without having to log on. Modem control is based on the assumption that the modem will, at least temporarily, drop the voltage on the Data Set Ready (DSR) line when the telephone connection is lost. It also assumes that the modem is correctly cabled such that a drop in DSR can be detected by the port.

The User Identification Code (UIC)

The User Identification Code (UIC) is a 32-bit number that identifies a user account. The username specified when a user logs on can be thought of as the name assigned to the UIC. In most places in the utility programs that a UIC is required as a parameter you can specify either the UIC itself or the username associated with the UIC.

The UIC is composed of two parts; the owner ID and the group ID. The owner ID is the most significant 16 bits of the UIC and the group ID is the least significant 16 bits.

When the UIC is displayed or typed in at the terminal, it is specified as two hexadecimal numbers, separated by a comma, enclosed in square brackets. Following is a list of sample UICs in the standard UIC syntax:

[1,1]	The UIC of the System account. It has an owner ID or 1 and a group ID of 1.
[0002,0007]	An owner ID of 2 and a group ID of 7.
[00bd,000f]	An owner ID of bd and a group ID of f.
[1001,3]	An owner ID of 1001 and a group ID of 3.
[3,1001]	An owner ID of 3 and a group ID of 1001.

The UIC Defines Ownership

There is a UIC associated with each resource defined in the system. The UIC specifies the ownership of that resource. For instance, if a file has a UIC of [2,7] then the file is owned by the user account whose UIC is [2,7].

There are two types of resources in the system: active resources, i.e., all processes or programs; and passive resources, i.e., files, devices, directories, and named memory segments. Active resources (processes) can potentially affect (modify) passive resources. Passive resources have no effect on active resources.

When a user successfully logs on to the system the LOGON program creates a process for that user, as specified in the User Authorization File (usually the process is CIP). The process created is owned (has the same UIC) by the user account associated with the username specified by the user. For instance, suppose the user logs on and specifies a username of "JANIS" and that the UIC assigned to the username JANIS is [3,15] (owner ID is 3 and group ID is 15). The LOGON program references the record in the user authorization file that belongs to JANIS to know which process to create, and which UIC to assign to the process. If the user passes any user validation (e.g. password) then LOGON creates a process. The UIC specifying the owner of that process is [3,15].

By default, all processes created by a process have the same UIC as the parent process. For instance, if CIP has a UIC of [3,15], and you execute another program (e.g. VEW), that other program will also have a UIC of [3,15]

By default, all files, directories and named memory segments created or defined by a process will be assigned the same UIC as the process that created them. For instance, a process with a UIC of [3,15] creates a file. That file will also have a UIC of [3,15]. Referencing (reading and/or writing) a file, device, directory, or named memory segment has no affect on ownership.

Directory type devices (disks and tapes) have a fixed ownership specified in the boot block or volume label. When a disk or tape device is mounted, the boot block or volume label is read, and the ownership is assigned according to the value stored there.

Non-directory type devices (tty ports, the Null device, pipes, etc.) do not have a fixed ownership. When one of these devices is mounted, the ownership is assigned as the UIC of the process that mounted it.

How To Display and Assign Ownership

The following sections describe how to display and assign ownership of processes, files, directories, devices and named memory segments. This is done using a combination of standard WMCS utility programs. Each of these utility programs is described in detail in the WMCS User Reference Manual. The following discussion only illustrates the use of these commands in the limited context of displaying and modifying ownership.

Note that assigning the ownership of any object requires operator privilege and either group or world privilege (defined below). If you do not have these privileges, you cannot change the UIC of an object. Typically, if you need to change the ownership of something, you should log on to the system account, which has all privileges, and then modify the ownership.

Why would you want to reassign the ownership of an object? Occasionally ownership is assigned improperly. For instance, suppose that you had an account on your machine for ROBERT. Robert leaves your organization, and you want to remove his account. He left several files on your machine. Since the files contain information that you need, you want to reassign the ownership of ROBERT's files to some other current system user.

Or suppose Bill initializes a diskette on his system and writes a file to it. He then brings the diskette to your system to allow you to make a copy of the file. Bill's UIC on his system is [3,16]. On your machine, there is no user defined with that UIC. You may want to modify the ownership of the diskette to pertain to one of your system users.

Processes

The PSTAT utility is used to display and set the ownership of processes. For instance, suppose you type this command:

```
> pstat :owner :username
```

This kind of display would appear on your screen:

```
  PID       Process Name       Owner     Username
--------  ---------------- ----------- ---------

00010002  Que_Manager      [0001,0001] SYSTEM
00010001  Logflush         [0001,0001] SYSTEM
00010669  CIP_PAM          [0007,0002] PAM
000109E5  vew_PAM          [0007,0002] PAM
0001061C  CIP_MARY         [0004,0002] MARY
00010A09  CIP_VEW          [0007,0002] PAM
00010A0C  copy_MARY        [0004,0002] MARY
0001003D  memtest_sys$user [0001,0001] SYSTEM
```

The first two columns identify each active process in the system. The third column shows the UIC that corresponds to the process. The fourth column is the username assigned to that UIC.

To set the ownership of a process, type a command like the following:

```
> pstat cip_pam :owner=mary
```

The following message will appear on your screen:

```
00010669 CIP_PAM Altered.
```

Now type the following command:

```
pstat cip_pam :owner :username
```

The following display will appear on your screen:

```
  PID       Process Name       Owner     Username
--------  ---------------- ----------- ---------

00010669  CIP_PAM          [0004,0002] MARY
```

Note that the :owner switch causes the UIC of the process to be displayed while the :owner= switch causes the UIC of the process to be changed.

Note also that you can specify either a username or a UIC as the value of the :owner= switch. For instance, since the UIC for Mary is [0004,0002] you could have typed:

```
> pstat cip_pam :owner=[4,2]
```

Files and Directories

The COPY, DIR, and FSTAT utility programs are used to display and set the ownership of files and directories. (Remember that a directory is one kind of file. That is, operations that apply to files also apply to directories.)

For instance, suppose you typed this command:

```
> dir :owner :username
```

This kind of report would appear:

```
Directory listing of _DS0/SUMMARY/
File name             Owner       Username
-------------------  -----------  ---------
PROJ.MAY.23          [0007,0002]  PAM
BUDGET.MAY.5         [0004,0002]  MARY
REPORT.MAY.19        [0007,0002]  PAM
```

The filename is displayed in the left-most column. The second and third columns show the UIC and username of the owner of each file. To modify the ownership of a file type the following:

```
> fstat report.may :owner=mary
```

This message will appear on your screen:

```
_DS0/SUMMARY/REPORT.MAY.17 Altered.
```

Now type the following:

```
> dir :owner :username
```

This kind of report will appear on your screen:

```
Directory listing of _DS0/SUMMARY/
File name             Owner       Username
-------------------  -----------  ---------
PROJ.MAY.23          [0007,0002]  PAM
BUDGET.MAY.5         [0004,0002]  MARY
REPORT.MAY.19        [0004,0002]  MARY
```

Note again that ownership can be specified as either a username (mary in this case) or as a UIC. That is, you could have typed:

```
> fstat report.may :owner=[4,2]
```

As mentioned before, (by default) files created by a process will have the same UIC as the process that created them. When you copy a file you are creating a new file. Therefore, the COPY utility will assign ownership to the new file. For instance, suppose that your username was MARY (UIC=[4,2]). Type the following command:

```
> copy proj.may proj.jun
```

This message will appear on your screen:

```
_DS0/SUMMARY/PROJ.MAY to _DS0/SUMMARY/PROJ.JUN copied.
```

Now type the following:

```
> dir :owner :username
```

This kind of report will appear on your screen:

```
Directory listing of _DS0/SUMMARY/
File name             Owner     Username
------------------  ----------- --------
PROJ.JUN.1          [0004,0002] MARY
PROJ.MAY.23         [0007,0002] PAM
BUDGET.MAY.5        [0004,0002] MARY
REPORT.MAY.19       [0007,0002] PAM
```

Note that the new file is owned by MARY instead of PAM. Sometimes it is useful to copy files and preserve the ownership. That is, you want to copy a set of files, but do not want to change the ownership of the new files. For this purpose, the COPY utility has a :preserve= switch. Suppose that you are logged on as MARY (UIC=[4,2]) and you type the following:

```
> copy report.may report.jun :preserve=owner
```

This message will appear on your screen:

```
_DS0/SUMMARY/REPORT.MAY to _DS0/SUMMARY/REPORT.JUN copied.
```

Now type the following:

```
> dir :owner :username
```

This kind of report will appear on your screen:

```
Directory listing of _DS0/SUMMARY/
File name              Owner       Username
-----------------  -----------  ---------

PROJ.JUN.1          [0004,0002] MARY
PROJ.MAY.23         [0007,0002] PAM
BUDGET.MAY.5        [0004,0002] MARY
REPORT.JUN.1        [0007,0002] PAM
REPORT.MAY.19       [0007,0002] PAM
```

Note that the new file (REPORT.JUN.1) still belongs to PAM (UIC=[7,2]).

Devices

The BTUP and DSTAT utility programs are used to display and set the ownership of devices. For instance, type the following command to display the attributes of _ttl4:

```
> dstat _ttl4
```

This kind of report will appear on your screen:

```
_TTl4,6Al                                                 10-Apr-1985 10:03
Class        : TTY            Hard errors : 0             Driver ID    : $1001
Read oper    : 10161          Soft errors : 0             Block size   : 1
Write oper   : 370231         Num to retry: 0             Cur num dev  : 17
Owner        : [0007,0002]    Username    : PAM           Allocated    : No
Protection   : S:  RE,P:       ,G:DWRE,O:DWRE
Term type    : T7000          In char cnt : 0             Duplex       : Full
Baud rate    : 9600           In buf size : 64            Parity       : Disabled
Data width   : 8 Bit          Out char cnt: 0             Current col  : 0
Stop bits    : 1              Out buf size: 128           Host Sync    : Bell
Packet term  : NoCntrlChr        ControlC        ControlO        ControlX
ControlU        ControlZ        NoAutobaud      Broadcast       ExpandTabs
Mask8Bit        NoModemCtrl     NoRemote        XonXoff
```

Note the UIC and username displayed on the fifth line of the display. The DSTAT command can be used to modify the ownership of the device. The ownership, as set by DSTAT will persist until one of three things occurs:

1. The ownership is changed with a subsequent DSTAT command.

2. The device is dismounted. When the device is mounted again, it is assigned ownership based on either the ownership specified in the boot block/volume label or the same as the process which mounted the device (as described above).

3. There is no process accessing the device. At this point the ownership is set to the "default owner" (if one is defined) as discussed in the next chapter on system security and file protection.

For instance, you might type the following sequence of commands:

```
> dstat _tt14 :owner=MARY
> dstat _tt14
```

This kind of report will appear on your screen:

```
_TT14,6A1                                                    10-Apr-1985 10:03
Class        : TTY           Hard errors : 0                 Driver ID    : $1001
Read oper    : 11421         Soft errors : 0                 Block size   : 1
Write oper   : 395655        Num to retry: 0                 Cur num dev  : 17
Owner        : [0004,0002]   Username    : MARY              Allocated    : No
Protection   : S:  RE,P:      ,G:DWRE,O:DWRE
Term type    : T7000         In char cnt : 0                 Duplex       : Full
Baud rate    : 9600          In buf size : 64                Parity       : Disabled
Data width   : 8 Bit         Out char cnt: 0                 Current col  : 0
Stop bits    : 1             Out buf size: 128               Host Sync    : Bell
Packet term  : NoCntrlChr       ControlC        ControlO        ControlX
ControlU        ControlZ        NoAutobaud      Broadcast       ExpandTabs
Mask8Bit        NoModemCtrl     NoRemote        XonXoff
```

As before the value of the :owner= switch can be either a username or a UIC. For instance, you could have typed the following to change the ownership of the device to MARY:

```
> dstat _tt14 :owner=[4,2]
```

The BTUP Command can be used on disk devices to change the ownership of the device and to change the boot block on the disk. BTUP does not affect the ownership of the device as currently mounted. To have the new ownership take effect, you have to dismount and remount the device; or you can use the DSTAT Command to change current ownership without remounting the device.

For instance, suppose that there is a diskette mounted as _dx0. To display the ownership of the diskette type the following command:

```
> btup _dx0
```

This kind of report will appear on your screen:

```
_DX0
Devname   : DX               Protection: S:  RE,P:    ,G:DWRE,O:DWRE
Label     : Transfer         Volumeid  : 0     Created   : 05-Apr-1985 17:08:18
Owner     : [0007,0002]      Cache     : 30    Readahead
Username  : PAM              Usercache : 8     NoAutoflush
Numbsect  : 616              Sectorsz  : 1024  NoForcedWrite
Fcbsector: 302               Shiftcnt  : 10    Drivetype : FLOP09a
Ialloc    : 10               Alloc     : 10
```

Note the UIC and username displayed in the first column, the fourth and fifth lines.

To modify the ownership, as stored in the boot block, type:

```
> btup _dx0 :owner=mary
```

Now type the following:

```
> btup _dx0
```

The following kind of report will appear on your screen:

```
_DX0
Devname  : DX           Protection: S:  RE,P:    ,G:DWRE,O:DWRE
Label    : Transfer     Volumeid  : 0     Created   : 05-Apr-1985 17:08:18
Owner    : [0004,0002]  Cache     : 30    Readahead
Username : MARY         Usercache : 8     NoAutoflush
Numbsect : 616          Sectorsz  : 1024  NoForcedWrite
Fcbsector: 302          Shiftcnt  : 10    Drivetype : FLOP09a
Ialloc   : 10           Alloc     : 10
```

Type the following command:

```
> dstat _dx0
```

This kind of report will appear on your screen:

```
_DX0,0B0                                               11-Apr-1985 08:54
Class        : Disk        Hard errors : 0             Driver ID    : $300D
Read oper    : 12          Soft errors : 0             Block size   : 1024
Write oper   : 2           Num to retry: 5             Cur num dev  : 1
Owner        : [0007,0002] Username    : PAM           Allocated    : No
Protection   : S:  RE,P:    ,G:DWRE,O:DWRE
Inter ractor: 1            Num cylndrs : 77            Disk density: Double
Num IOPB's   : 16          Num heads   : 2             Seek direct : Forward
Num sectors  : 616         Sector/track: 4             Cache size   : 32
Drive type   : FLOP09a                 NoRAWverify     NoWriteProtect
```

Note that the ownership, as stored in the boot block was changed, but that the current owner of the mounted device has not changed. To have the system believe that MARY is now the owner, you need to dismount and remount the device, or use the :owner= switch of the DSTAT command.

For disk and tape devices, the ownership is originally assigned and stored in the boot block when the device is initialized. By default, the ownership is set the same as the ownership of the DINIT process. This, of course, can be overridden with the :owner= switch on the DINIT command line.

Named Memory Segments

The MSTAT Command is used to display and modify ownership of any named memory segments. For instance, type the following command:

```
> mstat common
```

This kind of report for the memory segment named COMMON will appear on your screen:

```
COMMON
    Ref count :     0   Size:       32       Owner    : [0001,0001]
    Protection: S:  RE,P:    ,G:DWRE,O:DWRE  Username: SYSTEM
    Status    : linger          linked
```

The UIC appears on the second line (Owner : [0001,0001]), and the username appears on the third line of the display (Username: SYSTEM).

To modify the ownership of a memory segment type the following command:

```
> mstat common :owner=mary
```

Now the display of the memory segment named COMMON looks like this:

```
COMMON
    Ref count :     0   Size:       32       Owner    : [0004,0002]
    Protection: S:  RE,P:    ,G:DWRE,O:DWRE  Username: MARY
    Status    : linger          linked
```

Relationships Between a Process and a Resource

The UIC defines a relationship between a process and a resource. This relationship is fundamental to protection. From the perspective of a resource (a file, another process, a device, etc.), processes belong to one of three classes: owner, group or public.

The relationship a process enjoys with respect to a resource, in conjunction with process privileges (discussed below) and protection masks (discussed in the next chapter) determines how or in what ways the process can affect the resource. For example, a process without privileges can hibernate another process only if the process belongs to the "owner" class with respect to the process being hibernated.

Following is a definition of each of the three classes:

1. Owner. Both the resource and the process belong to the same user account. That is, they have the same UIC. Both the owner ID and the group ID fields of the UIC for the process and the resource are the same. In this case the process is the owner of the resource.

2. Group. The process belongs to some other user in the same group as the owner of the resource. That is, the group ID of the UIC

of the resource is the same as the group ID of the UIC of the process, and the owner ID is different. The process belongs to the same group as the resource.

3. The group ID of the UIC of the process is not the same as the group ID of the resource. In this case there is no relationship between the process and the resource. From the perspective of the resource the process is a member of the general public.

Process UIC	Resource UIC	Process class	Comment
[0007,0002]	[0007,0002]	Owner	The UICs are the same
[0007,0002]	[0001,0002]	Group	The group ID's are the same
[0007,0002]	[0007,0003]	Public	The group ID's are different

Note that you have to compare the UICs of both the process and the resource to determine the relationship. You cannot determine relationship by examining the usernames.

UICs with Special Significance

There are two UICs that are reserved and cannot be assigned to user accounts. They are [0001,0001] and [0000,0001].

The UIC [0001,0001] is reserved for the system account. The system account uses the username SYSTEM. All WMCS release files and directories belong to this user account, and have this UIC. The system manager uses this account to perform system manager functions (editing system configuration files, assigning new accounts, etc.)

The UIC [0000,0001] is reserved to mean that the resource is Unowned. There must not be any user accounts in the UAF.DAT file with this UIC. Any resource that is unowned can be accessed by any process as though that process was the owner of the resource.

You may want to create a user account whose username is BACKUP. This user account would be used for doing system backups. The UIC for this account would be [0002,0001]. That is, it is in the same group as the SYSTEM account.

UIC Selection

When selecting UICs for users on your system, keep the following suggestions in mind:

1. You should begin by defining which groups you will have on the system. Each user of your system would belong in at least one of

these groups. On a small machine (with 1-10 users), you may have only two or three groups. For example:

Group ID	Description
1	System accounts (system manager account, backup)
2	Main user group. The majority of your users fit into this category.
3-F	Additional user groups.
10	Guest group. Any guest accounts (people who occasionally use your machine) fit in this group.

2. Do not structure your users into too many groups. If there are more groups than necessary, your users will have difficulty accessing files to which they should have legitimate access. You may find, after you have used your machine for a while, that a different organization of groups is necessary.

3. If there is a person who has accounts on several machines, it is wise to give him accounts with exactly the same UIC on all machines. This will greatly facilitate his ability to transfer his files from one machine to the other.

4. Don't forget to assign unique numbers for restricted access accounts.

5. If one person logically belongs in two or more separate groups you may want to give him more than one UIC. You could create several user accounts for that one individual. He logs on as JIMDOC when he is doing documentation and as JIMDEV when he needs to do development. Note that this could be a bit annoying to the user (he has to log off and log back on as a different user whenever he changes job function). You should do this only if there is some compelling need.

6. You may need to set up two or more user accounts that have the same UIC. This is not recommended because all of those users would own the files created by each other. They would not be able to protect their files from each other. The displayed owner will always be the username that occurs first in alphabetical sequence, not the "true" owner.

Process Privileges

The WMCS defines thirteen process privileges. A process privilege is permission, granted to a process, to perform specific, privileged kinds of operations -- operations that could, if abused, compromise system security.

Each user account in the User Authorization File (UAF.DAT) defines which privileges that user's processes will have. When a user logs on, a process is created for him (typically the CIP). That process is granted the privileges specified for that user in the user authorization file. By default, all processes created by the CIP are given the same privileges as the CIP that created it.

Most user accounts should require no privileges. A process with no privileges can do anything to any resource or process that it owns. It can access any file or device that it owns. It cannot affect processes of other system users or access their devices and files unless the owner of that resource has granted permission to do so. In short, a user account with no privileges is almost harmless. That user cannot damage the system or any other system user, provided all other accounts are properly protected.

A user account with ANY privilege has the opportunity to breach system security by accessing resources that do not belong to it. On the other hand, there are some functions that some users need to do, as part of their job description, that require privileges. Each privilege, "in the hands" of a knowledgeable system user, allows him to perform beneficial functions.

As the system manager it is your responsibility to assess the needs and abilities of your system users, and judiciously grant them the privileges that they need to get their jobs done, and at the same time, provide as much security as possible.

Definition of Process Privileges

The following is a discussion of each process privilege:

SETPRIV

SETPRIV (SET PRIVilege) grants the process permission to assign or acquire any privilege whatsoever. Any process with this privilege can obtain any other privilege merely for the asking. It can also assign any privilege to any process that it can affect. A process with this privilege can create subprocesses that have more privileges than it has.

This privilege is very dangerous. A user that has this privilege can do anything to the system. It is sometimes useful to grant this privilege to individuals that usually run without any privileges, but occasionally require extra privileges. You should only give this privilege to users with whom you would trust your most valuable possessions.

SYSTEM

SYSTEM privilege places the process in the category or class of processes known as "system" processes. This classification has to do with access to files and devices. For instance, the owner of a file can deny the public access to the file, but grant the system user limited access. Its meaning is clearer in the context of the protection mask described in the next chapter.

This privilege should be reserved for system accounts. The BACKUP account should have SYSTEM privilege. If the BACKUP account has SYSTEM privilege, then it can make backup copies of all files to which the owners of the files (and the directories leading to those files) have granted the system read permission. This way the user can control which of his files and directories are to be backed up.

READPHYS

READPHYS (READ PHYSical) privilege grants the process permission to perform physical read operations to devices and system memory. For instance, a process with this privilege can read physical sectors on the disk, bypassing the file system. (That really means that a process with this privilege can inspect and/or copy any file at all, regardless of file protection). Also, a process with this privilege can read physical memory addresses. By being able to inspect physical memory, a process can inspect the operating system itself, operating system control structures, and the memory of any process on the system.

Grant READPHYS privilege only to users who need to perform these kinds of operations.

WRITEPHYS

WRITEPHYS (WRITE PHYSical) privilege grants the process permission to perform physical write operations to devices and system memory. A process with this privilege can, for instance, write any sector on a disk. This gives the process the potential power to destroy files and even to destroy the file system. On the other hand, it allows the knowledgable user to correct problems in the file system.

A process with WRITEPHYS can also write to any area of system memory. It can modify the operating system, any operating system control structures, or the memory of any process on the system.

Grant WRITEPHYS privilege only to users who need to perform these kinds of operations. In general, any user who has WRITEPHYS privilege probably also needs READPHYS privilege.

SETPRIOR

SETPRIOR (SET PRIORity) privilege allows a process to assign a higher priority to any process which it can affect. It can also assign a higher timeslice value to any process it can affect. It can create subprocesses that have a higher priority, and/or higher timeslice values than the parent process. It can also go into real time mode. A process with this privilege has the capacity to "take over" or monopolize the machine. On the other hand, any process that requires realtime mode needs to have this privilege.

CHNGSUPER

CHNGSUPER (CHaNGe to SUPERvisor mode) privilege allows a process to change the 68000 processor mode to supervisor. Once in supervisor mode the process has access to all of physical memory, and it can execute "privileged" instructions. Any process with this privilege has the capability to take over the machine.

This privilege should be granted very sparingly. Only knowledgeable users writing system-type programs that need to run in supervisor mode need this privilege.

BYPASS

BYPASS privilege allows a process access to files, devices, and named memory segments regardless of file protection. The process can "bypass" file protection. A process with this privilege can inspect, modify, create, delete, mount, dismount, etc. any device, file, or named memory segment.

This privilege should be granted very sparingly. When you grant this privilege to a user, you have given that user access to all files on the system. File protection does not exist for that user.

OPERATOR

OPERATOR privilege is intended to provide a system manager with the permissions necessary to perform operator type functions. With operator privilege, a process can:

1. Set the system clock
2. Change the protection mask on any file or device
3. Flush the cache on any disk or tape device
4. Assign, remove or modify a logical name in the system logical name table
5. Install files or deinstall files
6. Mount a "special" (non-file structured) device
7. Mount devices when the device driver is in memory instead of on a disk file (using the _MEMMNT system call)
8. Mount devices for which the device driver has not been installed
9. Adjust the process priority scheduling ratio
10. Set the ownership of a process, device or file (if the process also has group or world privilege)
11. Set the process name of any process it can affect
12. Set the window size on a network virtual circuit
13. Define or undefine a rotor list

Any of your users who need to perform operator functions need to have this privilege. Note that since this privilege allows the user to modify the protection mask of an object, a process with this privilege can bypass protection.

ALTUIC

ALTUIC (ALTernate UIC) privilege allows a process to affect a resource if the owner of the image file from which the process was created has access to the resource. For instance, suppose a file has a UIC of [0007,0002]. The protection mask assigned to the file grants members of the same group read permission to the file. The public is not granted any permission to the file. (e.g.

S:RE,P:,G:RE,O:RWED See the next chapter for a definition of the protection mask). A process whose group ID is not 2 would normally not be able to read the file. If, however, the process has ALTUIC privilege, and the group ID of the UIC assigned to the image file from which the process was created is 2, the process is allowed to read the file.

ALTUIC is useful for some cases of installed files (see below). It rarely would be an advantage for a user account to be assigned this privilege.

WORLD

WORLD privilege allows a process to affect any other process in the system. For instance, a process with world privilege can kill any other process, hibernate any other process, or do anything else to any other process, regardless of the UIC of the process being affected. Also, a process with WORLD privilege that also has OPERATOR privilege can modify the UIC of any process, file, device, or named memory segment.

WORLD privilege should be granted to a user account very sparingly. A process with WORLD privilege can circumvent file protection by modifying the ownership so that the process is the owner of the file.

GROUP

GROUP privilege allows a process to affect any other process that has the same group ID as the process. It can kill, hibernate, etc. any process that has the same group ID as itself. If the process also has OPERATOR privilege it can modify the owner ID portion of the UIC of any process, file, device, or named memory segment that has the same group ID as itself.

GROUP privilege is very useful for groups of users that frequently have to access one another's processes.

NETWORK

NETWORK privilege allows a process to attempt to perform remote operations. This privilege allows process requests to leave a computer. It is up to the security mechanisms on the remote computer to determine if a request can enter that computer and be honored.

SETATTR

SETATTR (SET ATTRibutes) allows a process to change its attributes. There are six standard attributes and four user-defined attributes. The standard attributes are:

1. Swappable — The process may be swapped out by the SWAPPER program.

2. Prezeromem — Each page of memory is zeroed before it is allocated to the process.

3. Postzeromem — Each page of memory the process had allocated is zeroed after the process terminates or after the process is released for any reason, including swapping.

4. Desencrypt — Data sent to another node on the network is encrypted using the DES encryption algorithm.

5. Fastencrypt — Data sent to another node on the network is encrypted using a fast encryption algorithm.

6. Watchdog — The process may be killed by the WATCHDOG program after a period of inactivity.

The four user-defined attributes may be defined by the system manager and used by locally written applications.

How to Display and Set Process Privileges

The PSTAT system utility is used to display and set process privileges. The USERPROF system utility is used to modify the privileges assigned to a user account.

For instance, to find the list of privileges assigned to a currently active task, type a command line like the following:

```
> pstat cip_pam :privilege
```

The following kind of display will appear on your screen:

```
00010141  CIP_PAM
   Privilege:
              NOSETPRIV     NOSYSTEM      NOREADPHYS    NOWRITEPHYS
              NOSETPRIOR    NOCHNGSUPER   NOBYPASS      NOOPERATOR
              NOALTUIC      NOWORLD         GROUP
```

From this display you can see that this process has GROUP privilege. No other privileges are assigned.

To set additional privileges type a command line like the one below:

```
> pstat cip_pam :privilege=bypass,nogroup
```

Now type the following command:

```
> pstat cip_pam :privilege
```

The following kind of display will appear on your screen:

```
00010141  CIP_PAM
   Privilege:
              NOSETPRIV      NOSYSTEM       NOREADPHYS     NOWRITEPHYS
              NOSETPRIOR     NOCHNGSUPER     BYPASS        NOOPERATOR
              NOALTUIC       NOWORLD        NOGROUP
```

Read the command description of PSTAT in the WMCS User's Reference Manual for details on displaying and setting privileges.

Installed Files

Sometimes there is a need for a process to have additional privileges than those brought to bear by the user. For instance, one of the system utility programs is DM. It displays information regarding the usage of system memory. In order to get the information it needs, it must change to supervisor mode and access several of the system data structures. So, DM requires CHNGSUPER privilege in order to operate correctly. There are many users who would like to execute DM, but it would be unwise to grant all of those users CHNGSUPER privilege just so that they could execute DM. Even though the privilege to change to supervisor mode holds the potential for compromising system security, there is nothing that the user can do with DM to compromise security.

What is needed is a way to grant the DM process some inherent privileges. Then the user need not have the privilege because the process has it already.

This need is met in the WMCS via a process known as installed files. There is a utility program called INSTALL that is used to install files, and display a list of installed files. You can read the details of the INSTALL Command in the User Reference Manual. When a file is installed, it can be given privileges. Whenever a process is created from that file, the operating system grants the process the combination of privileges with which the file was installed, along with the privileges specified by the parent process.

To install a file, the requesting process must have operator privilege and must either have the privilege to be given the installed file, or must have SETPRIV privilege.

A file might be installed with fewer privileges than it requires to run successfully. In this case, the process invoking such a process is expected to make up the difference. For instance, the BTUP utility program is installed with readphys privilege. This privilege will allow any user with read access to a device to examine the boot block parameters of the device. To set any boot block parameter, however, the user must have write privilege to the device, and either be the owner of the device, or have writephys privilege. In other words, to set a parameter in the boot block the user must supply the extra privileges required.

There are several of the standard WMCS utility programs that are installed. These utility programs are installed as part of the boot process. The privileges assigned to these utility programs have been carefully specified to allow users with no privileges to do things that do not compromise system security. The system manager should not modify the list of utility programs that are installed or the list of privileges assigned to each. Doing so could open the door for violations of system protection.

The following is a list of the utilities that are installed, along with a list of the privileges each is assigned:

Utility	Privileges
btup	readphys
chkd	system
copy	setpriv
dinit	operator
dm	chngsuper
fstat	writephys
keygen	world,bypass
logflush	chngsuper
logon	all
mnt	setpriv
nstat	chngsuper
password	bypass
pstat	chngsuper
qprint	system
recover	bypass
send	bypass
sp	system
talkt	writephys
tcopy	readphys,operator
wibug	chngsuper

Device drivers must also be installed if you want users without privileges to mount devices using those drivers. This is a precautionary measure to prevent a user from mounting a device using an errant device driver. Installed device drivers do not need privileges. If a device driver is not installed, the user must have operator privilege to mount a device using that driver.

The list of device drivers to be installed appears in the file named SYS$DISK/SYSLIB/DEVICEUP.COM. This file is maintained by the system manager, and the system manager is responsible for controlling that list. By default, (as the system is initially loaded), all WICAT supported device drivers are installed.

Chapter 8

System Security and File Protection

The material in this chapter is for inexperienced and experienced system managers.

File protection (which encompasses directory protection, device protection, and named memory segment protection) specifies which users (processes) can access which resources (files, devices, etc.). The owner of the file (device, etc.) specifies the protection. The description of the access permissions assigned to a resource is represented as a protection mask. File protection is influenced by ownership (UICs) and process privileges described in the previous chapter. System managers need to understand ownership and process privileges before they can fully understand file protection.

Note that permissions to affect processes are not associated with a protection mask. WORLD privilege and GROUP privilege determine which processes can be affected by which other processes.

The Protection Mask

The file protection mask is composed of four fields that represent the kind of access allowed to the resource (file, device, etc.) by various classes of processes. The four fields are Owner, Group, Public and System. Each field contains data specifying the type of access allowed to the resource by a process in that class. For instance, the Owner field of the protection mask might specify that processes of the class "Owner" (with respect to this resource) are allowed to read and write, but not delete the resource.

The previous chapter defined the UIC as a basis for relationships between a process and the resource the process attempts to access. These relationships define three of the four classes of process. Three relationships were defined:

Owner - The process is the owner of the resource, i.e. the UIC of the process is the same as the UIC of the resource.

Group - The process belongs to the same group as the resource. That is, the group ID field of the UIC of the process is the same the group ID of the UIC of the resource.

Public - The process does not belong to the same group as the resource. That is, there is no relationship defined between the process and the resource.

The protection mask defines the fourth class of process known as a system process.

System - A system process is any process that has SYSTEM privilege, as defined in the previous chapter.

Permissions

For each process, the owner of the resource specifies the type of permission allowed by that process to the resource. There are four different permissions, or types of access. These types of access are Read, Write, Delete, and Execute.

For instance, the owner of a resource can specify that Group processes have Read permission, System processes have Read permission, Public processes do not have have any permissions, and Owner processes have Read, Write, Delete and Execute permissions.

Each type of permission has a different definition depending on the type of resource. For instance, Write permission to a file means something different than Write permission to a directory.

Non-directory files

The protection mask for a file is recorded in the File Control Block (FCB) for the files on the disk. For tape files it is recorded in the file header.

Read permission means that the process is authorized to open the file for read access, i.e., the file can be read by the process.

Write permission means that the process is authorized to write to the file (open the file for write access).

Execute permission applies to image files, and means that the process is authorized to execute the file, i.e., the user can create a process from the file.

Delete permission means that the process is authorized to delete the file, i.e., remove it from the file system.

Directory files

The protection mask for directory files is stored in the File Control Block (FCB) for the directory file. Note that there are no directory files on tape devices.

Read permission means that the process can inspect the contents of the directory, i.e. the names of the files that are contained in the directory. A process that has read privilege to a directory can change default to that directory, and list its contents. If the process has appropriate permissions to files contained in the directory it can open those files.

Write permission to directory files allows users to "write" to the directory, i.e. make modifications to the contents of the directory file. Operations that modify the contents of a directory include creating a file in the directory, deleting a file in the directory, and renaming a file from a directory to the same or another directory.

Execute permission to a directory file allows processes to traverse through the directory to access files in sub-directories. Execute permission does not allow a process to inspect the contents of a directory, or to open any files in the directory. For instance, to open a file named _dc0/direct.sub/file.ext.1 for read access, a process needs to have Read permission to the device (_dc0), Execute permission to all directories in the path leading to the directory containing the file (/rootdir/rootdir.dir and /rootdir/direct.dir in this case), Read permission to the directory containing the file (/direct/sub.dir in this case) and Read permission to the file itself.

Delete permission to a directory file authorizes the process to delete the directory file.

Devices

For non-directory devices (e.g. terminal ports) there is no permanent protection mask. When a non-directory device is mounted it is assigned the default protection mask of the process that mounts it. For directory devices (disk and tape) the protection mask is stored in the boot block or volume label for the device.

Read permission to a device allows a process to read from the device. For non-directory type devices, the meaning of this is intuitive. For directory devices (disk and tape) a process must have Read permission to a device before it can inspect any of the contents of the device, regardless of the protection mask associated with any of the contents of the device.

Write permission to a device allows a process to write to or make modifications on the device. For non-directory devices (e.g. terminal ports) the meaning is intuitive. For directory devices (disk and tape) a process must have Write permission to the device before it can modify any of the contents of the device, regardless of the protection mask associated with any of the contents of the device.

Execute permission on a device is undefined.

Delete permission to a device allows a process to dismount or mount the device.

Named Memory Segments

Read permission allows a process to share the named memory segment but it does not allow the process to make modifications to it. However, if the memory segment contains program code, the process can execute it.

Write permission allows a process to share the memory segment and read or write to it. The process can also execute program code in the memory segment.

Execute permission on a named memory segment is undefined.

Delete permission allows a process to delete the named memory segment.

Syntax of the Protection Mask

The file protection mask is displayed and specified in the following format:

```
S:DWRE,P:DWRE,G:DWRE,O:DWRE
```

What each letter represents:

S	System processes
P	Public processes
G	Group processes
O	Owner processes
D	Delete permission
W	Write permission
R	Read permission
E	Execute permission

Note that the class of process (i.e., system, public, group, or owner) is specified, followed by a colon and the letters indicating the type of permission granted to that class of process. Each class is separated from the next by a comma. When specifying a protection mask, the order in which the classes of processes are specified and the order in which the permissions are specified is arbitrary. A protection mask may not contain spaces.

When specifying a protection mask you need only specify the fields that you want to modify. For instance, if you want to keep the currently defined permissions for Owner and Group processes, but want to change the permissions for the System and Public processes, you need only specify values for the System and Public fields.

Consider the following examples:

```
g:rew,p:,o:rwed,s:re
```

Group processes have Read, Execute and Write permission. Public processes have no permissions. Owner processes have all permissions. System processes have both Read and Execute permissions.

```
p:re,s:re
```

Permissions for Group and Owner processes are left unmodified. Public and System processes are granted Read and Execute permissions.

```
o:dewr
```

Permissions for Group, Public and System processes are left unmodified. Owner processes are granted all permissions.

```
g:,s:,p:re
```

Permissions for Owner processes are left unmodified. Group and System processes have no permissions. Public processes are granted Read and Execute permissions.

How to Display and Set the Protection Mask

Different commands are used to display and set the protection mask on files and directories, devices, and named memory segments.

Files and Directories

The DIR Command is used to display the protection mask assigned to files and directories. For instance, consider the following command:

```
> dir :protection

Directory listing of _DS0/SUMMARY/
File name                Protection
------------------ -------------------------

PROJ.MAY.23          S:  RE,P:    ,G:DWRE,O:DWRE
BUDGET.MAY.5         S:  RE,P:    ,G:DWRE,O:DWRE
REPORT.MAY.21        S:  RE,P:    ,G:DWRE,O:DWRE
```

The first column lists the names of files found in the directory named _DS0/SUMMARY/. The second column shows the protection mask assigned to each file. Note that when the protection mask is displayed, spaces are used to align the various fields. Spaces, however, may not be used when you specify a protection mask.

The FSTAT Command is used to modify the protection mask assigned to a file or directory. Consider the following:

```
> fstat budget.may :protection=p:re,s:d
_DS0/SUMMARY/BUDGET.MAY Altered.

> dir :protection

Directory listing of _DS0/SUMMARY/
File name                Protection
------------------ -------------------------

PROJ.MAY.23          S:  RE,P:    ,G:DWRE,O:DWRE
BUDGET.MAY.5         S:D   ,P:  RE,G:DWRE,O:DWRE
REPORT.MAY.21        S:  RE,P:    ,G:DWRE,O:DWRE
```

Note that only the two fields specified (System and Public) were modified. The permissions specified for Group and Owner processes were not affected.

Note also that the value specified replaced the previous value. For instance, before the change, System processes were granted Read and Execute permissions. The FSTAT Command specified Delete permission for System processes. After the change, System processes did not have Read and Execute permissions. The previously specified permissions were replaced.

When a file is copied there are three things considered in determining what protection mask is assigned to the new file: the default protection mask, the :preserve= switch, and the :protection= switch.

By default, the file is assigned the default protection mask associated with the process. (The default protection mask is discussed below). For instance, suppose the default protection mask associated with the current process is S:RE,P:RE,G:RE,O:DWRE, and the following commands were executed:

```
> copy budget.may budget.jun
_DS0/SUMMARY/BUDGET.MAY to _DS0/SUMMARY/BUDGET.JUN copied.

> dir :protection

Directory listing of _DS0/SUMMARY/
File name                    Protection
-------------------  --------------------------
PROJ.MAY.23          S:  RE,P:     ,G:DWRE,O:DWRE
BUDGET.JUN.1         S:  RE,P:  RE,G:  RE,O:DWRE
BUDGET.MAY.5         S:D   ,P:  RE,G:DWRE,O:DWRE
REPORT.MAY.21        S:  RE,P:     ,G:DWRE,O:DWRE
```

Note that the new file received the default protection mask. If you would like to preserve the protection associated with the original file, issue the following commands:

```
> copy budget.may budget.jun :preserve=protection
_DS0/SUMMARY/BUDGET.MAY to _DS0/SUMMARY/BUDGET.JUN copied.

> dir :protection

Directory listing of _DS0/SUMMARY/
File name                    Protection
-------------------  --------------------------
PROJ.MAY.23          S:  RE,P:     ,G:DWRE,O:DWRE
BUDGET.JUN.1         S:D   ,P:  RE,G:DWRE,O:DWRE
BUDGET.MAY.5         S:D   ,P:  RE,G:DWRE,O:DWRE
REPORT.MAY.21        S:  RE,P:     ,G:DWRE,O:DWRE
```

Note that with the :preserve=protection switch the newly created file is assigned the same protection mask as the original file.

If you like, you can specify the protection mask to be applied to the new file on the COPY Command line. Note that if you do not specify a value for all four fields, the default protection mask is used to assign a value to unspecified fields.

Consider the following:

```
> copy budget.may budget.jun :protection=p:,s:rw
_DS0/SUMMARY/BUDGET.MAY to _DS0/SUMMARY/BUDGET.JUN copied.

> dir :protection

Directory listing of _DS0/SUMMARY/
File name                 Protection
-------------------  --------------------------

PROJ.MAY.23          S:  RE,P:    ,G:DWRE,O:DWRE
BUDGET.JUN.1         S: WR ,P:    ,G:  RE,O:DWRE
BUDGET.MAY.5         S:D   ,P:  RE,G:DWRE,O:DWRE
REPORT.MAY.21        S:  RE,P:    ,G:DWRE,O:DWRE
```

Note that values from the default protection mask were used for the Group and Owner fields. The System and Public fields were assigned the specified values.

The VEW utility program preserves the protection mask for files that are edited.

Devices

The DSTAT and BTUP utility programs are used to display and modify the protection mask for devices. For instance, consider the following command:

```
> dstat _tt14

_TT14,6A1                                               12-Apr-1985 10:21
Class        : TTY            Hard errors : 0           Driver ID    : $1001
Read oper    : 29288          Soft errors : 0           Block size   : 1
Write oper   : 1131931        Num to retry: 0           Cur num dev  : 17
Owner        : [0007,0002]    Username    : TLS         Allocated    : No
Protection   : S:  RE,P:    ,G:DWRE,O:DWRE
Term type    : T7000          In char cnt : 0           Duplex       : Full
Baud rate    : 9600           In buf size : 64          Parity       : Disabled
Data width   : 8 Bit          Out char cnt: 0           Current col  : 0
Stop bits    : 1              Out buf size: 128         Host Sync    : Bell
Packet term  : NoCntrlChr         ControlC          ControlO          ControlX
ControlU       ControlZ           NoAutobaud        Broadcast         ExpandTabs
Mask8Bit       NoModemCtrl        NoRemote          XonXoff
```

Note the protection mask on the sixth line of the display. The DSTAT Command can be used to modify the protection mask. The protection mask assigned with DSTAT persists until one of three things occurs:

1. The protection mask is changed with a subsequent DSTAT Command.

2. The device is dismounted. When the device is mounted again, it is assigned a protection mask based on either the protection mask in the boot block/volume label or the default protection mask associated with the process that mounts the device.

3. There is no process accessing the device. At this point the protection mask is set to the "default device protection mask" (not to be confused with the default protection mask).

For instance, consider the following:

```
> dstat _ttl4 :protection=p:re
_TTl4 Altered.
> dstat _ttl4

_TTl4,6Al                                           12-Apr-1985 10:21
Class       : TTY         Hard errors : 0           Driver ID    : $1001
Read oper   : 29288       Soft errors : 0           Block size   : 1
Write oper  : 1131931     Num to retry: 0           Cur num dev  : 17
Owner       : [0007,0002] Username    : TLS         Allocated    : No
Protection  : S:  RE,P:  RE,G:DWRE,O:DWRE
Term type   : T7000       In char cnt : 0           Duplex       : Full
Baud rate   : 9600        In buf size : 64          Parity       : Disabled
Data width  : 8 Bit       Out char cnt: 0           Current col  : 0
Stop bits   : 1           Out buf size: 128         Host Sync    : Bell
Packet term : NoCntrlChr      ControlC       Control0       ControlX
ControlU      ControlZ        NoAutobaud     Broadcast      ExpandTabs
Mask8Bit      NoModemCtrl     NoRemote       XonXoff
```

Note that since only the Public field was specified, it was the only field that changed.

The BTUP Command can be used on disk devices to change the protection mask of the device. Note that using BTUP to modify the protection mask of a disk device changes the boot block on the disk. It does not affect the ownership of the device as currently mounted. To have the new protection mask take effect, you have to dismount and remount the device.

For instance, suppose that there is a diskette mounted as _dx0. To display the protection mask of the diskette, type the following command:

```
> btup _dx0
```

The following kind of display appears on your screen:

```
DX0
Devname  : DX             Protection: S:  RE,P:    ,G:DWRE,O:DWRE
Label    : Transfer       Volumeid  : 0     Created   : 05-Apr-1985 17:08:18
Owner    : [0007,0002]    Cache     : 30    Readahead
Username : PAM            Usercache : 8     NoAutoflush
Numbsect : 616            Sectorsz  : 1024  NoForcedWrite
Fcbsector: 302            Shiftcnt  : 10    Drivetype : FLOP09a
Ialloc   : 10             Alloc     : 10
```

Note the protection mask displayed on the second line of the display.

To modify the protection mask as stored in the boot block, type:

```
> btup _dx0 :protection=p:re
```

Now type the following command:

```
> btup _dx0
```

A display like the following appears on your screen:

```
DX0
Devname  : DX             Protection: S:  RE,P:  RE,G:DWRE,O:DWRE
Label    : Transfer       Volumeid  : 0     Created   : 05-Apr-1985 17:08:18
Owner    : [0007,0002]    Cache     : 30    Readahead
Username : PAM            Usercache : 8     NoAutoflush
Numbsect : 616            Sectorsz  : 1024  NoForcedWrite
Fcbsector: 302            Shiftcnt  : 10    Drivetype : FLOP09a
Ialloc   : 10             Alloc     : 10
```

For disk and tape devices, the protection is originally assigned and stored in the boot block when the device is initialized. By default, the protection is set to the default protection mask associated with the DINIT process. This, of course can be overridden with the :protection= switch on the DINIT Command line.

Named Memory Segments

The MSTAT Command is used to display and modify the protection mask of any named memory segments.

For instance, consider the following:

```
> mstat common
```

```
COMMON
    Ref count :      0  Size:       32      Owner    : [0001,0001]
    Protection: S:  RE,P:    ,G:DWRE,O:DWRE  Username: SYSTEM
    Status    : linger          linked
```

To modify the protection mask of a memory segment type the following command:

```
> mstat common :protection=p:re
```

Now the display for the memory segment named COMMON looks like this:

```
> mstat common

COMMON
    Ref count :      0   Size:        32        Owner    : [0001,0001]
    Protection: S:  RE,P:  RE,G:DWRE,O:DWRE   Username: SYSTEM
    Status    : linger              linked
```

Note that since only the Public field was specified in the :protection= switch, it was the only field that changed.

The Default Protection Mask

The default protection mask is a protection mask associated with a process. It does not define permissions for accessing the process. Rather, it is used as a default value for files created (devices mounted, named memory segments defined) by that process. When a process creates a subprocess, the subprocess inherits the same default protection mask that was associated with the parent process.

In the User Authorization File there is a default protection mask assigned to each user account. When a user logs on, LOGON creates a process (typically CIP) for that user. It obtains the default protection mask for that user account from the UAF.DAT file, and assigns it to the process being created.

The OPTION Command can be used to display and set the default protection mask. For instance, consider the following:

```
     > option

     Home         : SYS$DISK/PAM/
     Log          : Yes
     Message      : Yes
     Path         : ,/sysexe/,/sysexe.users/
     Pause        : No
     Prompt       : PAM>
     Protection : S:  RE,P:    ,G:DWRE,O:DWRE
     SYS$RESULT : 0
     Username     : PAM
     Verify       : No
     Trace        : No
```

The default protection mask is displayed on the seventh line of the display.

To modify the default protection mask, type:

```
     > option :protection=p:re
```

Now type the following:

```
     > option
```

The following kind of display appears on your screen:

```
     Home         : SYS$DISK/PAM/
     Log          : Yes
     Message      : Yes
     Path         : ,/sysexe/,/sysexe.users/
     Pause        : No
     Prompt       : PAM>
     Protection : S:  RE,P:  RE,G:DWRE,O:DWRE
     SYS$RESULT : 0
     Username     : PAM
     Verify       : No
     Trace        : No
```

A user may want to specify a default protection mask in his USERUP command file so that a specific protection mask will be assigned each time he logs on. (You should only have to do this if you are dissatisfied with the default protection mask assigned to your user account by the system manager in the User Authorization File). To do this, insert a command like the following into the USERUP.COM file:

```
    > option :protection=s:re,p:re,g:re,o:dwre :perm
```

Note that the :perm switch is required for this default protection mask to persist beyond the scope of the USERUP command file.

Process Privileges and File Protection

Several of the process privileges directly and/or indirectly affect file protection. They are summarized here as they apply to file protection. For a complete description of process privileges, see chapter 7 in this manual. As can be seen in this summary, a process with any privilege (except SETPRIOR) can circumvent file protection to one degree or another.

ALTUIC A process with this privilege can access files as though it had the UIC of the owner of the image file from which the process was created. For instance, all of the WMCS utility programs are owned by [0001,0001]. If a process has ALTUIC privilege and was created from one of the files in /SYSEXE/ it can access files for which [0001,0001] has access.

BYPASS A process with this privilege bypasses all file protection checking. Nothing can be hidden from a process with this privilege.

CHNGSUPER A process with this privilege can change to supervisor mode of operation and can, by writing to its process control block, give itself BYPASS privilege, (change its UIC, ...) and then it can access any file on the system.

GROUP A process with this privilege in combination with OPERATOR privilege can change its UIC to any UIC in the same GROUP, and thus become the owner of any file owned by another user in the same group. (The owner of a file can change the protection mask of that file.)

OPERATOR A process with this privilege can assign the protection mask of any resource. It need not own the resource to assign the protection mask. If the process also has WORLD privilege, it can assign the UIC (owner) of any resource. If the process has both OPERATOR and GROUP privileges it can assign the UIC (owner) of any resource owned by any other user in the same group.

READPHYS A process with this privilege can read physical sectors from the disk. By doing this, the process can read any file from the disk, without having been granted access to the file by the owner.

SETPRIV A process with this privilege can grant itself other privileges such as BYPASS or READPHYS, and then it can access any file on the system.

SYSTEM A process with this privilege is in the system class and has permission to access files as defined in the system portion of the protection mask.

WORLD A process with this privilege in combination with OPERATOR privilege can change its UIC to any UIC, and thus become the owner of any file. (The owner of a file can change the protection mask of that file).

WRITEPHYS A process with this privilege can write physical sectors on the disk. By doing this, it can write to any file on the disk without having been granted access to the file by the owner.

Device Ownership and Protection

Each device has a default UIC and a default protection mask assigned to it. The LOGON program can control access to devices.

LOGON

Before allowing a user to log on, the LOGON program checks if the user has both read and write privilege to the specified device. It takes into account the UIC that corresponds to the username specified, the privileges that correspond to the username, and the protection mask assigned to the port. If the user cannot both read and write the port, LOGON fails and does not create the process specified in the UAF.DAT file for that user.

If LOGON succeeds, it reassigns the UIC of the port to the UIC of the user logging on. This way, the user logging on owns the port and can assign any of the port characteristics without needing any process privileges. If LOGON did not change the ownership of the port, the user logging on would not be able to assign port characteristics.

Default Ownership and Protection of Devices

Assigned to each device is a default UIC and a default protection mask. (This is not to be confused with the default protection mask associated with each process.) When the device is idle, i.e. not opened by any process, the ownership and protection mask of the device are set to the default value.

For instance, suppose that you are logged on to _tt14 and that your UIC is [0007,0002]. When you logged on, the LOGON utility assigned the UIC of _tt14 to be your UIC. As long as you are logged on, even though you are not constantly typing characters, _tt14 is not idle. The CIP has _tt14 open, waiting for any command. However, when you log off, the port becomes idle and the ownership (UIC) and protection mask are reset to the "default" values.

The reason for this is clear. Suppose user A logs on to a port. The ownership of that port is assigned to user A. User A modifies the protection mask so that only he has read and write privilege to the device. Then user A logs off. If the UIC and protection mask did not go back to a predefined default value, no one other than user A (or a user with BYPASS privilege) would be able to log on at that port.

In most situations, the system manager sets up the default UIC and protection mask for each device so that any user can log on at any terminal port. To do this the default UIC is set to the "Unowned" UIC [0000,0001]. Remember that if the UIC of a resource is [0,1] that any process can access the device as though it were the owner.

If you wanted to restrict logons at a certain terminal, you could set the default UIC and protection so that only a particular user or a group of users would be able to log on.

There is no way to display the default UIC or protection mask assigned to a device. To set the default UIC and/or protection use the DSTAT Command. For example, type:

```
     > dstat _tt14 :owner=[0,1] :default
```

This command assigns the default UIC of _tt14 to [0000,0001].

The default ownership and/or protection are typically assigned in the DEVICEUP.COM file. The system manager edits this file and sets the default ownership and protection as needed on his system. A typical excerpt from the DEVICEUP.COM file is:

```
! Port: pl  Username/function:
            mnt _tt9
            dstat _tt9 :termtype=t7000 :expandtabs :owner=[0,1] :default
```

Note that ownership is set to the Unowned UIC with the :default switch.

Organizing for File Protection

Organizing files carefully can enhance the file protection capabilities. For instance, suppose that Charles has files scattered across several directories on the system. For Charles to do his work successfully he needs access to all of the directories that contain his files. If other users also have files in those same directories, then the protection on those directories has to be sufficiently lenient to all users who need access to them. Even though all users may not be able to read one another's files, they at least know of their existence. (All users would have to have Read privilege to the directory containing their files. Read privilege to the directory allows them to see the names of all files in the directory, regardless of the protection on the file itself.)

On the other hand, if all of Charles' files are in a single main directory and its subdirectories, then access to those files can be controlled easily· by restricting access to not only the files themselves, but also to the directories containing the files.

This can be carried one step further. If there is more than one disk on your system you can divide your users such that some users have their files on one disk, and some on the other. Then you can control access to the files by setting up protection on the disks so that users only have access to the disk that contains their files.

In general, it is a good idea to give each user his own directory. That user completely controls the files in that directory. If he owns the directory file itself, he can set protection on both the directory file and on the files within the directory.

Additionally, each group of users on your system might have a set of directories that only members of the group can access.

By keeping files organized in a controlled set of directories, the owners of the files can control access to their files. If files are haphazardly scattered across several directories, controlling access to the files is difficult at best.

Trouble-shooting File Protection

Occasions arise where a user cannot access a device or file that he legitimately needs to access. When this happens you need to discover the source of the problem and correct it. The problem could be:

1. The user does not have the correct permissions for the device.

2. The user does not have the correct permissions for one or more of the directories leading to the file.

3. The user does not have the correct permissions for the file itself.

The following example illustrates how to discover where the problem is. Suppose that user Terri (whose username is TERRI) comes to you and says that she cannot access the file named _DS0/USERS.GROUP1.TEST/SYSTEM.DOC. The following steps illustrate the solution:

Step 1 | Find out TERRI's UIC and process privileges by using the PSTAT command. (Remember that file protection is based on ownership and process privileges.) Type the following:

```
> pstat :owner :username :privilege :uic=terri
```

This kind of display appears on your screen:

```
00011D1A  CIP_TERRI          [0003,0002] TERRI
    Privilege:
                NOSETPRIV      NOSYSTEM       NOREADPHYS     NOWRITEPHYS
                NOSETPRIOR     NOCHNGSUPER    NOBYPASS       NOOPERATOR
                NOALTUIC       NOWORLD        NOGROUP
```

From this display you learn that Terri's UIC is [3,2], and that she has no privileges. The main privileges that you watch for are SYSTEM and BYPASS. If the user has SYSTEM privilege, then when you look at protection masks, you inspect the field for system processes. If the user has BYPASS privilege, then there is no reason they cannot access a file.

Step 2 | Find out the owner and protection mask for the device by using the DSTAT Command. Type the following:

```
> dstat _ds0
```

This kind of display appears on your screen:

```
_DS0,0A0                                          12-Apr-1985 10:21
Class        : Disk          Hard errors : 0      Driver ID    : $300B
Read oper    : 720837        Soft errors : 7      Block size   : 1024
Write oper   : 189106        Num to retry: 5      Cur num dev  : 1
Owner        : [0001,0001] Username    : SYSTEM   Allocated    : No
Protection   : S: WRE,P: WRE,G: WRE,O:DWRE
Inter factor: 3              Num cylndrs : 842    Disk density: Double
Num IOPB's   : 32            Num heads   : 20     Seek direct : Forward
Num sectors : 421000         Sector/track: 25     Cache size   : 152
Drive type   : SMD474b                NoRAWverify NoWriteProtect
```

Note the UIC of the owner of this volume is [1,1]. Now you must determine which field of the protection mask applies to this user. To do this, ask yourself these questions:

> Is the UIC of the device the same as the UIC of the user process? If the answer is yes, look at the Owner field of the protection mask.
>
> Is the group ID of the UIC of the device the same as the group ID of the UIC of the user process? If the answer is yes, look at the Group field of the protection mask.
>
> Does the process have SYSTEM privilege? If the answer is yes, look at the System field of the protection mask.
>
> What is in the Public field of the protection mask?

Note that more than one of the fields may apply. For instance, if the group IDs are the same, you need to look at both the Group field and the Public field. If the UICs are the same, look at the Owner field, the Group field and the Public field. If the user process has access to the device in ANY of the categories that apply, the user process has access to the device.

In this case, the UICs don't match ([0001,0001] does not match [0003,0002]). The group IDs do not match (the group ID of the device is 0001 while the group ID of the process is 0002). The process does not have SYSTEM privilege, so the only category that applies is Public.

The protection mask for this device is: "S: WRE,P: WRE,G: WRE,O:DWRE"

Note that the Public field contains "P: WRE" meaning that the Public has Write, Read and Execute permissions for the device. In fact, the only permission not granted to the Public is Delete permission. If you recall, Delete permission allows the process to mount or dismount the device, and Execute permission on a device is undefined. Since Terri was trying to access a file on the device, she needs only Read and possibly Write permission to the device. This she has.

Step 3 | Check the permissions on all of the directory files involved.

To access a file, the user process must have execute privilege to all directories in the path leading to the directory that contains the file. In this case there are three directories in the path leading to the directory that contains the file:

```
_DS0/ROOTDIR/ROOTDIR.DIR.1
_DS0/ROOTDIR/USERS.DIR.1
_DS0/USERS/GROUP1.DIR.1
```

Use the DIR command to inspect the ownership and protection mask assigned to each of these directory files.

```
> dir rootdir/rootdir.dir :owner :username :protection

Directory listing of _DS0/ROOTDIR/
File name              Owner     Username         Protection
-------------------  ----------- ---------  ------------------------------
ROOTDIR.DIR.1        [0001,0001] SYSTEM     S:  RE,P:  RE,G: WRE,O: WRE
```

Ask yourself the same questions as before (in step 2) to determine which field of the protection mask to look at. In the case of /ROOTDIR/ the correct field is Public because the user does not have System privilege, the UICs do not match, and the group IDs do not match.

In this case the Public has Read and Execute permissions. So the problem is not with this directory.

```
> dir rootdir/users.dir :owner :username :protection

Directory listing of _DS0/ROOTDIR/
File name              Owner     Username         Protection
-------------------  ----------- ---------  ------------------------------
USERS.DIR.1          [0001,0001] SYSTEM     S:  RE,P:  RE,G: WRE,O: WRE
```

The problem is not with this directory, either. Terri, in this case again, is a member of the Public, and the Public has Execute permission as required.

```
> dir users/groupl.dir :owner :username :protection
```

```
Directory listing of _DS0/USERS/
File name              Owner     Username        Protection
-----------------   -----------  --------  -------------------------
GROUP1.DIR.1        [0001,0002]  WILEY     S:  RE,P:  RE,G: WRE,O: WRE
```

In this directory, Terri is a member of the Group (the owner of the directory [1,2] has the same group ID as the owner of the user process [3,2]). So in this case you look at both the Public field and the Group field of the protection mask. In both cases, Terri's user process has Execute permission, so the trouble is not with this directory.

The user process must also have at least Read, and possibly Write permission to the directory containing the file. If the process only wants to affect that single file, Read permission to the directory containing the file is sufficient. If, however, the process also wants to create a new file in that directory (e.g. VEW creates temporary files, and a new version of the destination file in the directory) the process must also have Write permission to the directory.

```
> dir users.groupl/test.dir :owner :username :protection
```

```
Directory listing of _DS0/USERS.GROUP1/
File name              Owner     Username        Protection
-----------------   -----------  --------  -------------------------
TEST.DIR.1          [0001,0002]  WILEY     S:  RE,P:  RE,G:  RE,O: WRE
```

Again, this file belongs to a member of Terri's group (the group ID of Terri and Wiley are the same). Look at both the Group field and the Public field of the protection mask. In both cases, Terri is given Read and Execute permission.

But Terri was attempting to VEW the file named _DS0/USERS.GROUP1.TEST/SYSTEM.DOC, and since VEW creates files in the directory, it requires Write permission to this directory. You have found the problem.

Step 4 | Give Terri the Write permission she needs to this directory file with the FSTAT command. Type the following:

```
> fstat users.groupl/test.dir :protection=g:wre
```

Step 5 | Use the DIR command to check the protection mask. Type the following:

```
> dir users.groupl/test.dir :owner :username :protection
```

This kind of display appears on your screen:

```
Directory listing of _DS0/USERS.GROUP1/
File name               Owner     Username          Protection
------------------   -----------  ---------  ------------------------------
TEST.DIR.1           [0001,0002]  WILEY      S:  RE,P:  RE,G: WRE,O: WRE
```

Step 6 | You may have solved the problem with this one change. Just to be sure, check the file itself with the DIR command. Type the following:

```
> dir users.group1.test/system.doc :owner :username :protection
```

This kind of display appears on your screen:

```
Directory listing of _DS0/USERS.GROUP1.TEST/
File name               Owner     Username          Protection
------------------   -----------  ---------  ------------------------------
SYSTEM.DOC.12        [0003,0002]  TERRI      S:  RE,P:  RE,G:  RE,O: WRE
```

Since the file belongs to Terri, look at the Owner, Group, and Public fields of the protection mask. Since Terri is trying to VEW the file, she needs Read permission. This she has. Now Terri can have access to the file.

This same procedure will work for any file you are trouble-shooting.

CHAPTER 9

INITIALIZATION OF MEDIA

This chapter is written for experienced system managers. Do not concern yourself with this chapter if you are new to the WMCS.

9.1 CIP COMMANDS RELATED TO INITIALIZING MEDIA

The following commands pertain to media initialization (read the WICAT Multi-user Control System (WMCS) User Reference Manual for information on how to execute these commands):

BTUP	Update the boot block on a disk
DINIT	Initialize a disk or diskette

See the chapter in this manual on How to Monitor and Customize System Use for information on some aspects of system tuning affected by the BTUP and DINIT commands.

9.2 DISKS AND DISKETTES

A disk is initialized, i.e., ready for the WMCS to use it, when the following things have been done to the disk:

1. The disk's tracks and sectors have been formatted so that the device driver knows the layout, i.e., the format, of the disk.

2. A boot block has been written to the disk or diskette at sector zero.

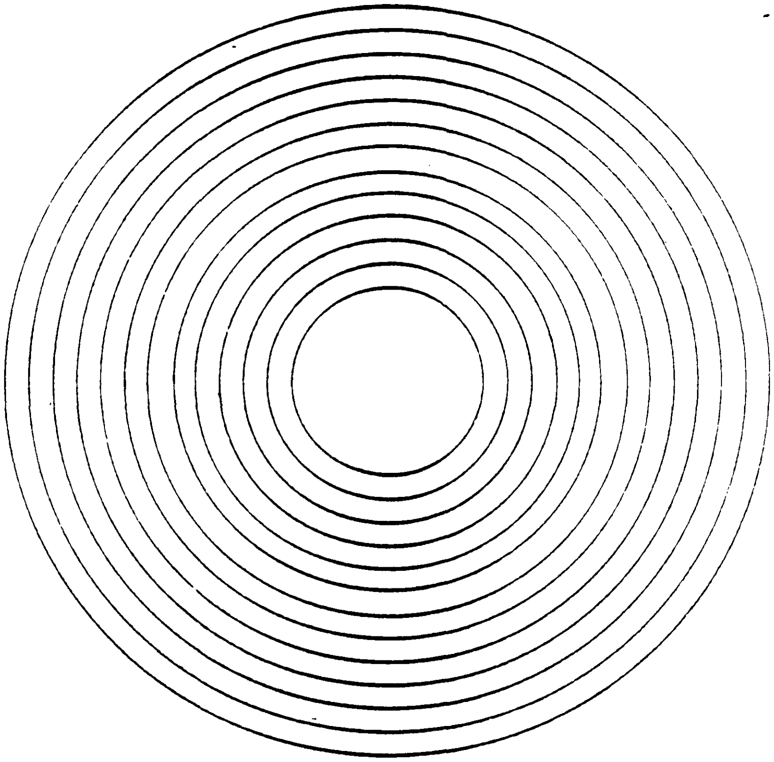
3. The following files have been created on the volume:

   FCB.SYS
   ROOTDIR.DIR
   FCBBITMAP.SYS
   BITMAP.SYS

The following sections explain what each of these steps involves.

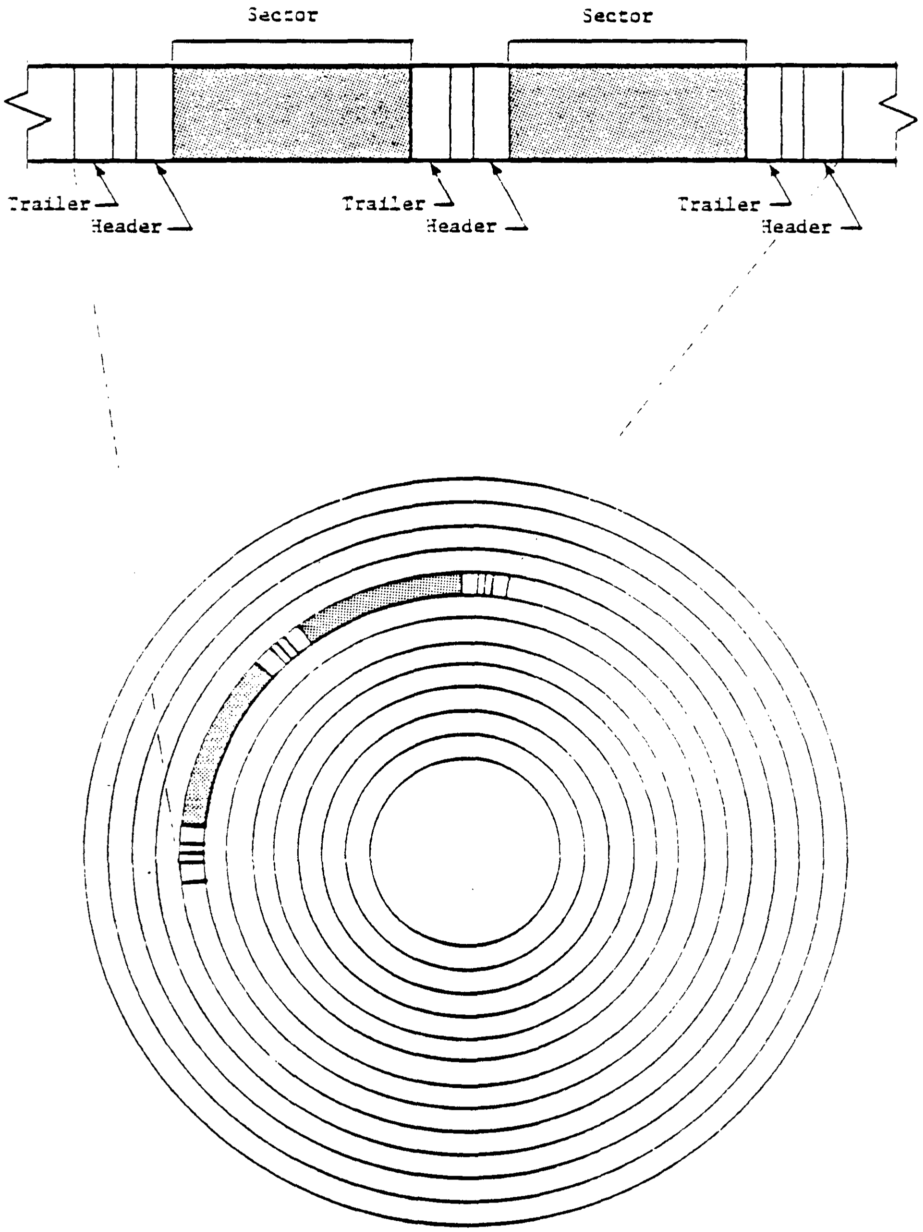
9.3 FORMATTING

The tracks on a disk are concentric rings about the center of the disk:



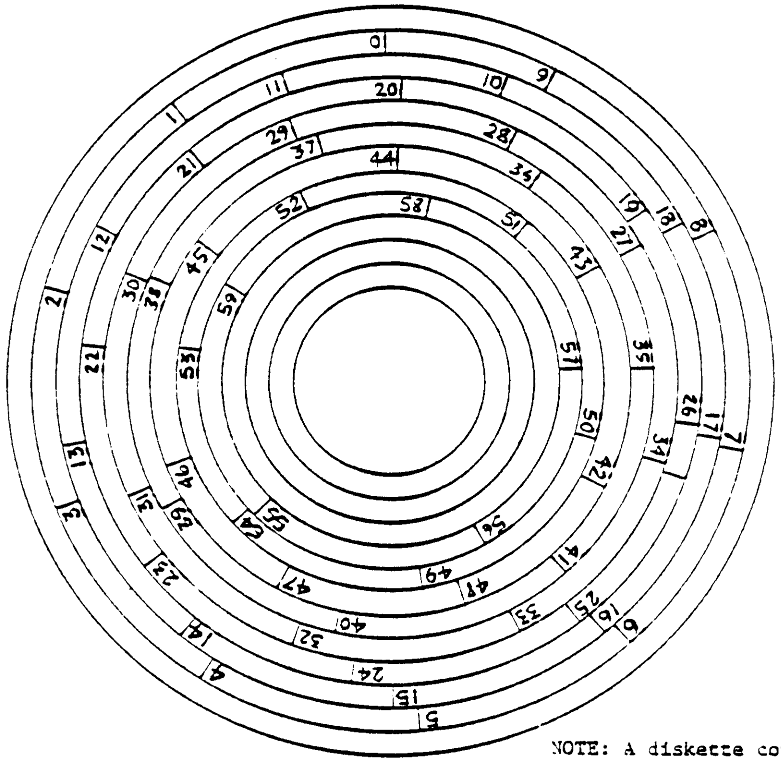
NOTE: A diskette contains many more tracks than the number pictured here.

When the WMCS formats a disk, it first places the disk drive's head at the beginning of the first track on each surface of the disk. The WMCS then subdivides each track into sectors by writing a <u>sector header</u> and a <u>sector trailer</u>, at regular intervals, until all the space in all the tracks on each disk surface is filled. Each header and trailer define a sector.



NOTE: A diskette contains many more tracks than the number pictured here.

Once the sectors have been laid out. the WMCS assigns a number to each sector, starting with 0 for the first sector and continuing until a number. or address, is assigned to every sector.



NOTE: A diskette contains many more tracks than the number pictured here.

Formatting must be performed the first time a disk is initialized for use under the WMCS. Once formatted, however, the volume need not be reformatted unless it has behaved eratically, e.g., consistent difficulty in reading files from the disk, writing files to the disk, etc.

9.4 WRITING THE BOOT BLOCK TO THE DISK

Once the disk is formatted, the CIP command used to initialize a disk then creates a boot block and writes it to sector zero on the disk being initialized.

This is what the boot block looks like when you display it on a terminal screen:

```
00000000  55 54 49 4C 49 54 49 45 53 20 34 2E 31 2E 30 00  UTILITIES 4.1.0.
00000010  07 BE 01 10 0C 02 02 0E 00 01 00 01 00 10 FF FF  ................
00000020  00 01 00 01 00 01 04 00 00 0A 00 0A 00 02 00 00  ................
00000030  02 68 44 58 30 00 00 00 00 00 00 00 01 2F 00 7F  .hDX0......../..
00000040  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000050  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000060  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000070  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000080  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000A0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000B0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000C0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000D0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000E0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000F0  00 00 00 00 00 00 00 00 00 00 00 00 52 D4 25 C6  ............R.%.
```

Line numbers

Each line in the display is assigned a number for the user's convenience.

Bit field

These 16-byte lines contain the hexadecimal-value equivalences for the characters on the same line in the column on the right of the display.

ASCII field

This column contains the ASCII-character equivalences to the values on the same line in the bit field.

Read the description of the ZAP Command in the WMCS user reference manual for more information on the format of the display.

The following information tells you what the boot block contains:

```
00000000  55 54 49 4C 49 54 49 45 53 20 34 2E 31 2E 30 00  UTILITIES 4.1.0.
00000010  07 BE 01 10 0C 02 02 0E 00 01 00 01 00 10 FF FF  ................
00000020  00 01 00 01 00 01 04 00 00 0A 00 0A 00 02 00 00  ................
00000030  02 68 44 58 30 00 00 00 00 00 00 00 01 2F 00 7F  .hDX0......../..
00000040  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000050  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000060  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000070  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000080  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000A0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000B0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000C0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000D0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000E0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000F0  00 00 00 00 00 00 00 00 00 00 00 52 D4 25 C6  ............R.ž.
```

Volume label

This is the label that the user assigns to the disk. It appears on the screen when a disk drive, containing the volume, is mounted.

```
00000000  55 54 49 4C 49 54 49 45 53 20 34 2E 31 2E 30 00  UTILITIES 4.1.0.
00000010  07 BE 01 10 0C 02 02 0E 00 01 00 01 00 10 FF FF  ................
00000020  00 01 00 01 00 01 04 00 00 0A 00 0A 00 02 00 00  ................
00000030  02 68 44 58 30 00 00 00 00 00 00 01 2F 00 7F  .hDX0......../..
00000040  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000050  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000060  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000070  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000080  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000A0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000B0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000C0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000D0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000E0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000F0  00 00 00 00 00 00 00 00 00 00 00 00 52 D4 25 C6  ............R.ž.
```

Volume initialization date

Volume ID number

User Identification Code (UIC)

This is the UIC of the user who initialized the volume.

Protection mask

This is the protection assigned to the volume. In other words, this field tells you the privileges, regarding the volume, that are assigned to each class of users.

These are the classes into which users on your system are divided:

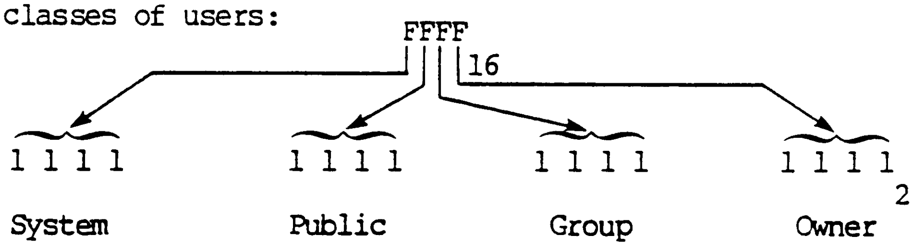
Owner — The user who created the file, i.e., initialized the volume.

Group — The group to which the owner is assigned.

Public — Users who are not members of the owner's group.

System — The SYSTEM user(s), i.e., processes that have SYSTEM privilege.

Each hexadecimal digit in this field represents one of the foregoing classes of users:



For each class of users, there are four things that can be done with a volume:

Delete — The volume can be mounted and dismounted.

Write — The contents of the volume can be modified.

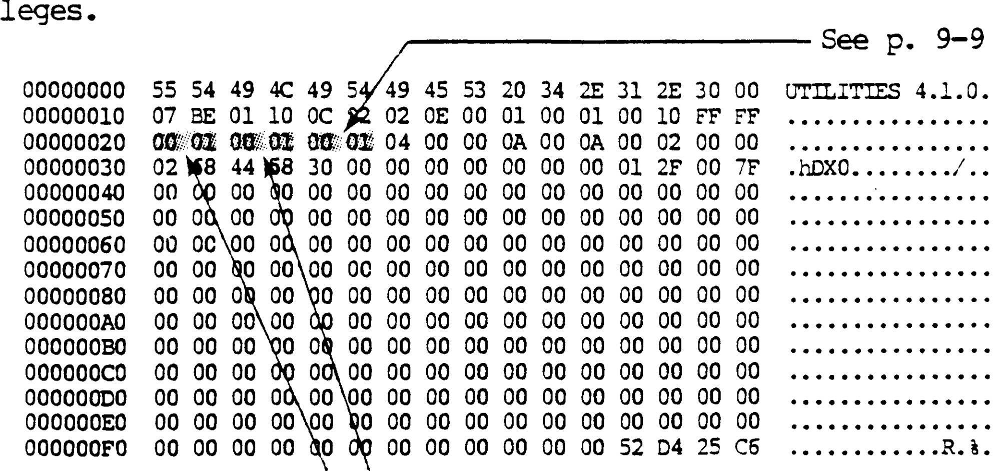
Read — The contents of the volume can be perused.

Execute — Undefined.

Therefore. a single value in each digit in this field of the boot block tells you the privileges assigned to the users whose class is represented by that digit. The following chart tells you how to interpret the values that can appear in each digit. In other words, one of the values in the lefthand column will appear in each of the four digits in this field of the FCB. The first value in the File Protection field of the FCB tells you the privileges the system user has, the second tells you the privileges the public, has, etc. Thus. if an F appears for the first digit. you know that the system user can delete, write, read, and execute the volume (a one in any of the four columns next to the Volume Protection Value indicates that the privilege to perform the function represented by that column is granted). If an A appears for that first digit, the system can only delete and read the volume.

Volume Protection Value	Delete	Write	Read	Execute
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
A	1	0	1	0
B	1	0	1	1
C	1	1	0	0
D	1	1	0	1
E	1	1	1	0
F	1	1	1	1

For example, F2 6F as the value appearing in this field of the boot block indicates that the system user has all privileges regarding the volume, that the public can only read the volume, the group can write and read the volume, and the owner has all privileges.



See p. 9-9

```
00000000  55 54 49 4C 49 54 49 45 53 20 34 2E 31 2E 30 00  UTILITIES 4.1.0.
00000010  07 BE 01 10 0C 92 02 0E 00 01 00 01 00 10 FF FF  ................
00000020  00 01 00 01 00 01 04 00 00 0A 00 0A 00 02 00 00  ................
00000030  02 68 44 58 30 00 00 00 00 00 00 01 2F 00 7F     .hDX0......../..
00000040  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000050  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000060  00 0C 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000070  00 00 00 00 00 0C 00 00 00 00 00 00 00 00 00 00  ................
00000080  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000A0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000B0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000C0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000D0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000E0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000F0  00 00 00 00 00 00 00 00 00 00 00 00 52 D4 25 C6  ............R.ì.
```

Initial sector-allocation

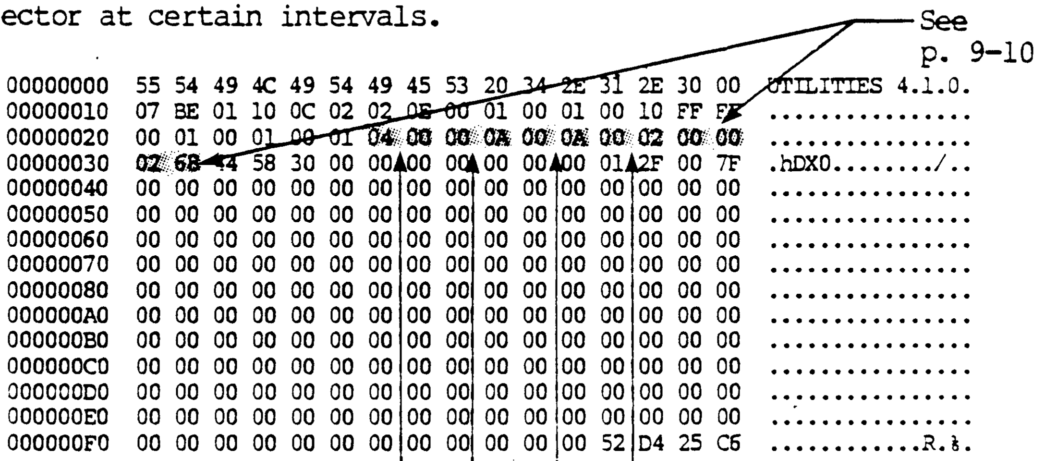
This value corresponds to the parameter value assigned, when the disk was initialized (or when the boot block was last updated), to specify the number of sectors on the disk that will be allocated to a file when a file is created.

Subsequent sector-allocation

This value corresponds to the parameter value assigned, when the disk was initialized (or when the boot block was last updated), to specify the number of sectors on the disk that will be allocated to a file each time that file grows beyond the sectors already allocated to it.

Autoflush flag

This value corresponds to the parameter value assigned, when the disk was initialized (or when the boot block was last updated), to specify whether the data in memory that are assigned to a sector on the disk will automatically be written back to that disk sector at certain intervals.



See p. 9-10

```
00000000  55 54 49 4C 49 54 49 45 53 20 34 2E 31 2E 30 00  UTILITIES 4.1.0.
00000010  07 BE 01 10 0C 02 02 0E 00 01 00 01 00 10 FF FF  ................
00000020  00 01 00 01 00 01 04 00 00 0A 00 0A 00 02 00 00  ................
00000030  02 68 44 58 30 00 00 00 00 00 00 00 01 2F 00 7F  .hDX0......../..
00000040  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000050  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000060  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000070  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000080  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000A0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000B0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000C0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000D0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000E0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000F0  00 00 00 00 00 00 00 00 00 00 00 00 00 52 D4 25 C6  ............R.ì.
```

Sector size

This is the size (in bytes) of the sectors on the disk.

Shift Count

This tells you the number of bit positions (to the left) that the one-bit must be moved to equal the number of bytes in each sector on the volume, e.g., ten for a sector size of 1024 bytes, 9 for a 512-byte sector size.

Disk-cache size

This value corresponds to the parameter value assigned, when the disk was initialized (or when the boot block was last updated), to specify the size of the cache set aside in memory for data from sectors on the disk, i.e., the amount of memory that is reserved (when a device containing this disk is mounted) for data from the disk.

User-cache size

This value corresponds to the parameter value assigned, when the disk was initialized (or when the boot block was last updated), to specify the maximum number of sectors that any process can request at one time from the disk.

The total number of sectors on the disk

```
00000000  55 54 49 4C 49 54 49 45 53 20 34 2E 31 2E 30 00  UTILITIES 4.1.0.
00000010  07 BE 01 10 0C 02 02 0E 00 01 00 01 00 10 FF FF  ................
00000020  00 01 00 01 00 01 04 00 00 0A 00 0A 00 02 00 00  ................
00000030  02 68 44 58 30 00 00 00 00 00 00 00 01 2F 00 7F  .hDX0......../..
00000040  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000050  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000060  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000070  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000080  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000A0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000B0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000C0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000D0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000E0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000F0  00 00 00 00 00 00 00 00 00 00 00 00 52 D4 25 C6  ............R.ì.
```

Boot-disk name

This is the name (assigned by the user when the bootblock was lasted modified) by which the disk (not the device in which the disk may be mounted) is known if the disk is used to boot the system.

Address of FCB.SYS

This is the address on the disk at which the FCB.SYS file begins.

Mounted flag

This value indicates whether or not the device in which the disk is located is mounted.

```
00000000  55 54 49 4C 49 54 49 45 53 20 34 2E 31 2E 30 00  UTILITIES 4.1.0.
00000010  07 BE 01 10 0C 02 02 0E 00 01 00 01 00 10 FF FF  ................
00000020  00 01 00 01 00 01 04 00 00 0A 00 0A 00 02 00 00  ................
00000030  02 68 44 58 30 00 00 00 00 00 00 00 01 2F 00 7F  .hDX0......../..
00000040  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000050  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000060  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000070  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000080  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000A0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000B0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000C0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000D0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000E0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000F0  00 00 00 00 00 00 00 00 00 00 00 00 52 D4 25 C6  ............R.ȝ.
```

Readahead

This value corresponds to the parameter value assigned, when the disk was initialized (or when the boot block was last updated), to specify whether or not the WMCS assumes that if a process reads sectors from the disk sequentially, the WMCS should be ready (for any sector requested) to read the next logically sequential sector from the disk.

Reserved

These are reserved for enhancement of the WMCS.

Boot-block checksum

This is the checksum for the values assigned to each byte in the boot block.

9.5 WRITING STANDARD FILES TO THE DISK

When the boot block has been written to sector zero, the disk initialization program (the CIP command used to initialize disks) creates the following files on the disk (in the order indicated):

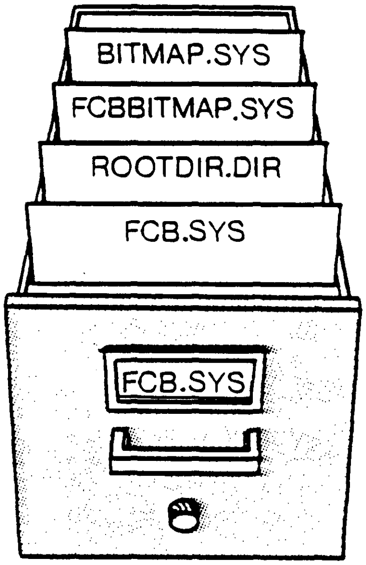
FCB.SYS
ROOTDIR DIR
FCBBITMAP.SYS
BITMAP.SYS

The following sections describe each of these files.

9.5.1 FCB.SYS

FCB.SYS contains a record for each file on the disk. Each record is called a File Control Block (FCB) and contains information that the WMCS uses to locate the file assigned to the FCB and provide information on the file as requested, e.g., the DIR Command.

FCB.SYS is therefore the system file (hence the .SYS file extension) that contains an FCB for each file on a particular disk. Were you to think of FCB.SYS as a card file drawer, this is what it would contain upon completion of disk initialization:



Were you to display the contents of FCB.SYS on a terminal screen. each FCB would look something like this:

```
00000100   00 00 00 01  00 01 00 00  00 00 00 00  01 01 00 01   ................
00000110   52 4F 4F 54  44 49 52 00  00 44 49 52  00 01 00 00   ROOTDIR..DIR....
00000120   00 01 00 01  00 20 00 01  00 01 FF FF  07 BE 01 12   ..... .......>..
00000130   0B 20 37 3A  07 BE 01 12  0B 20 37 3A  00 00 00 00   . 7:.>... 7:....
00000140   00 00 04 00  00 00 04 00  00 00 00 01  00 00 01 2C   ...............,
00000150   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000160   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000170   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000180   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000190   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001A0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001B0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001C0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001D0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001E0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001F0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 DD D4   ..............]T
```

Line numbers

These line numbers are for your convenience. and indicate the relative position (within the file) of the first byte on the line.

Bit field

Each line in this fields of the display contains the hexadecimal values corresponding to the ASCII characters on the same line in the column on the righthand side of the display.

ASCII field

The characters on each line in this field correspond to the hexadecimal values on the same line in the display's bit field.

The following information tells you what each field of the FCB contains:

```
00000100   00 00 00 01  00 01 00 00  00 00 00 00  01 01 00 01   ................
00000110   52 4F 4F 54  44 49 52 00  00 44 49 52  00 01 00 00   ROOTDIR..DIR....
00000120   00 01 00 01  00 20 00 01  00 01 FF FF  07 BE 01 12   ..... .......>..
00000130   0B 20 37 3A  07 BE 01 12  0B 20 37 3A  00 00 00 00   . 7:.>... 7:....
00000140   00 00 04 00  00 00 04 00  00 00 00 01  00 00 01 2C   ...............,
00000150   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000160   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000170   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000180   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000190   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001A0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001B0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001C0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001D0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001E0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001F0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 DD D4   ..............]T
```

FCB number

Each FCB in FCB.SYS is assigned a serial number (the series reflects the order in which the FCBs in the file were created). Inasmuch as each FCB is a record, the FCB number serves as the record number.

For example, when a diskette is initialized, FCB.SYS is the first file the disk initialization program creates on the diskette, and approximately 50 FCBs are created in FCB.SYS.

Inasmuch as FCB.SYS is the first file created on a disk, FCB 0 is assigned to FCB.SYS. FCB 1 is assigned to ROOTDIR.DIR. the second file created on a disk; FCB 2 is assigned to FCBBITMAP.SYS; and FCB 3 is assigned to BITMAP.SYS.

The remaining, unassigned, FCBs are left blank until other files are created on the disk, e.g., FCB 4 is assigned to the fifth file created on the disk, etc. As the number of unassigned FCBs is exhausted, new FCBs are created automatically.

Note that when a file is deleted from the disk, the FCB assigned to that file is not deleted. For example, suppose that you initialize a disk and then copy a file named STATUS.DOC onto that disk; STATUS.DOC is assigned to FCB 5. After copying the file to the disk you realize that you copied the wrong file. You delete STATUS.DOC from the disk and then copy ANALYSIS.DOC to the disk.

When you delete STATUS.DOC, the WMCS writes a value in one of the fields in FCB 5 that indicates that FCB 5 is unallocated, i.e., available for assignment (this field is described later in this chapter). Therefore, when you copy ANALYSIS.DOC to the disk, and the WMCS searches FCB.SYS to find the first available (unassigned) FCB, it finds FCB 5 and assigns that FCB to ANALYSIS.DOC.

Therefore. the FCB number pertains to the creation of the FCB and has nothing to do with the creation or deletion of the file to which it is assigned; it remains constant once the FCB is created.

```
00000100   00 00 00 01  00 01 00 00  00 00 00 00  01 01 00 01   ................
00000110   52 4F 4F 54  44 49 52 00  00 44 49 52  00 01 00 00   ROOTDIR..DIR....
00000120   00 01 00 01  00 20 00 01  00 01 FF FF  07 BE 01 12   ..... .......>..
00000130   0B 20 37 3A  07 BE 01 12  0B 20 37 3A  00 00 00 00   . 7:.>... 7:....
00000140   00 00 04 00  00 00 04 00  00 00 00 01  00 00 01 2C   ...............,
00000150   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000160   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000170   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000180   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000190   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001A0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001B0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001C0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001D0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001E0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001F0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 DD D4   ..............]T
```

FCB Sequence Number

You may have noticed that when a file is created, the sequence number assigned to its FCB (the number appearing under SEO in a directory listing) frequently matches the number of the FCB, e.g., the FCB.SEQ number is 1458.1458.

The sequence number is used primarily to uniquely identify the FCB. The sequence number is also incremented by one each time a file is assigned to the FCB.

Recall the example, from the foregoing section. concerning FCB 5 and files STATUS.DOC and ANALYSIS.DOC that were assigned to it. When FCB 5 was assigned to STATUS.DOC, it was the first time that FCB 5 had been assigned to a file. Therefore. the FCB Sequence Number for FCB 5 might have been 00 05. Then. when ANALYSIS.DOC was assigned to FCB 5, the FCB Sequence Number would have been one greater than the original sequence number.

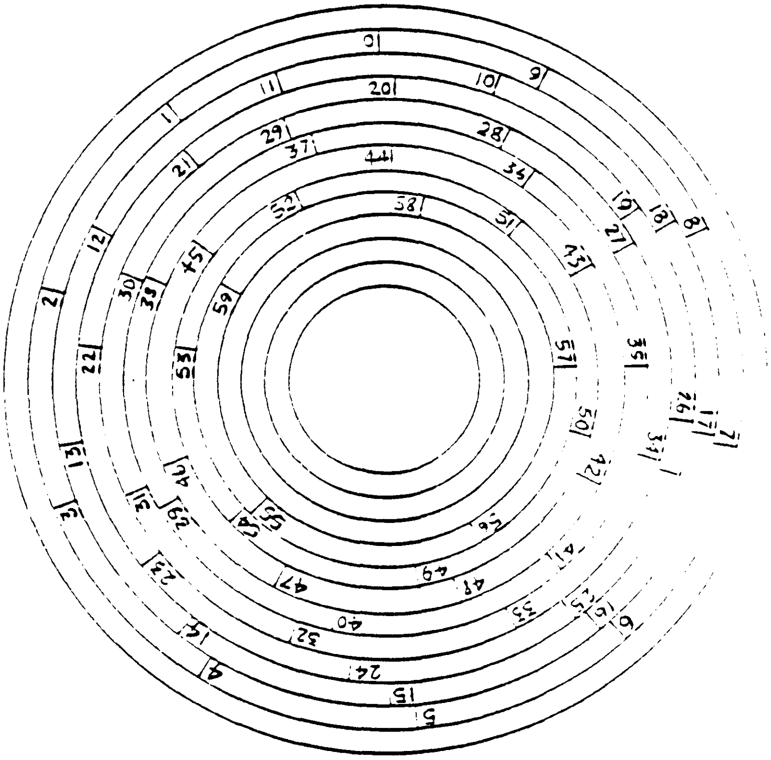
```
00000100   00 00 00 01  00 01 00 00  00 00 00 00  01 01 00 01   ................
00000110   52 4F 4F 54  44 49 52 00  00 44 49 52  00 01 00 00   ROOTDIR..DIR....
00000120   00 01 00 01  00 20 00 01  C0 01 FF FF  07 BE 01 12   ..... .......>..
00000130   0B 20 37 3A  07 BE 01 12  0B 20 37 3A  00 00 00 00   . 7:.>... 7:....
00000140   00 00 04 00  00 00 04 00  00 00 00 01  00 00 01 2C   ...............,
00000150   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000160   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000170   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000180   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000190   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001A0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001B0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001C0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001D0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001E0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001F0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 DD D4   ..............]T
```

Continuation FCB Number

This field of the FCB contains information about one aspect of the physical location of the data that constitute the file assigned to the FCB.

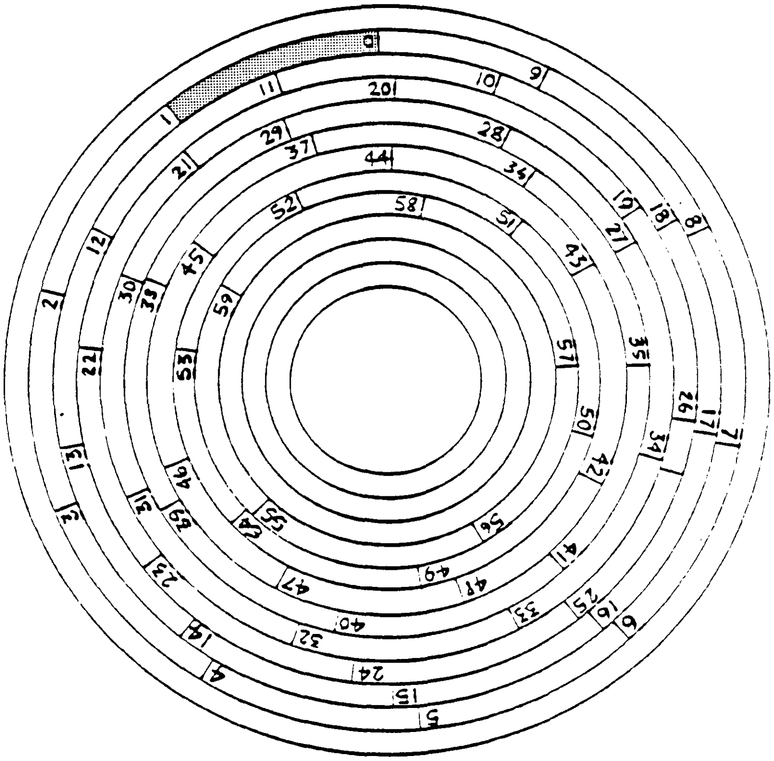
When this field is filled with 0s, no continuation FCBs have been assigned to the FCB.

The data written to a disk are assigned to sectors on that disk. For example, this is what a formatted disk might look like before any data has been written to it:



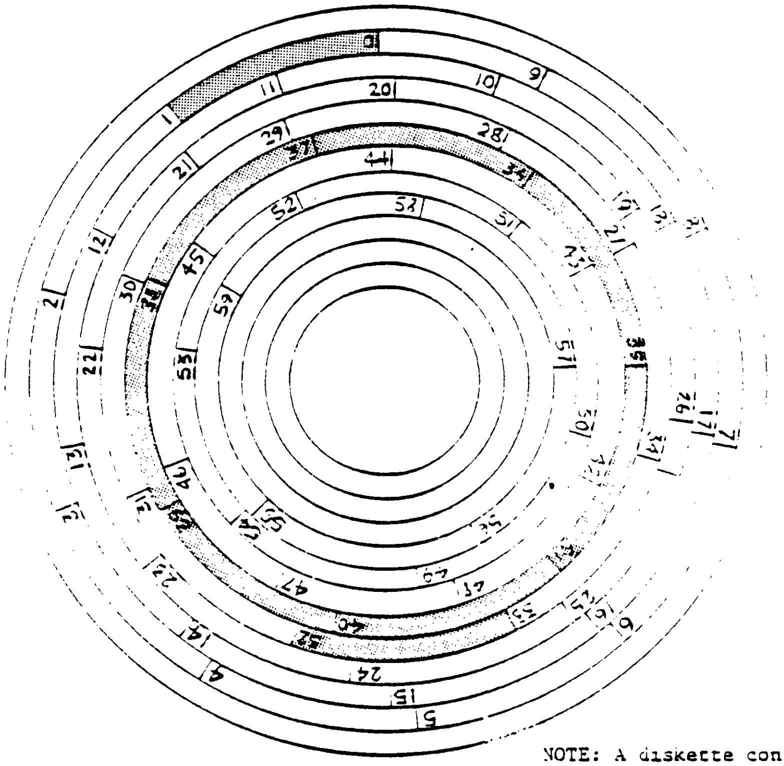
NOTE: A diskette contains many more tracks than the number pictured here.

This is what the same disk might look like when the boot block has been written to sector zero (shading indicates that data have been written to the sector):



NOTE: A diskette contains many more tracks than the number pictured here.

This is what the same disk might look like when initialization is complete, i.e., the data constituting the four standard files have also been written to the volume:



NOTE: A diskette contains many more tracks than the number pictured here.

Now, suppose that you want to copy a single file to the disk pictured in the foregoing illustration, and that that file is large enough to fill every available sector on the volume. When the WMCS writes the file to the disk, part of the file is written to the first group of available sectors, and the other part to the second group of sectors. This means that the data constituting the file are contained in two groups of contiguous sectors. Each group of contiguous sectors is an <u>extent</u>.

A file can comprise any number of extents, but there is room on an FCB for only 30 extents; that is, room for information on the location and size of each sector assigned to the file. Therefore, when a file consists of more than 30 extents, another FCB must be assigned to the file as a <u>Continuation FCB</u>. The Continuation FCB Number field of the FCB contains the FCB number of the Continuation FCB assigned to the file (if it has been necessary to assign a Continuation FCB).

Therefore, a value of 00 00 00 08 in this field of the FCB indicates that a Continuation FCB has been assigned to accommodate additional sectors, and that that FCB is number 008 in FCB.SYS.

```
00000100   00 00 00 01  00 01 00 00  00 00 00 00  01 01 00 01   ................
00000110   52 4F 4F 54  44 49 52 00  00 44 49 52  00 01 00 00   ROOTDIR..DIR....
00000120   00 01 00 01  00 20 00 01  00 01 FF FF  07 BE 01 12   ..... .......>..
00000130   0B 20 37 3A  07 BE 01 12  0B 20 37 3A  00 00 00 00   . 7:.>... 7:....
00000140   00 00 04 00  00 00 04 00  00 00 00 01  00 00 01 2C   ...............,
00000150   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000160   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000170   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000180   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000190   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001A0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001B0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001C0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001D0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001E0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001F0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 DD D4   ..............]T
```

Continuation FCB Sequence Number

This field contains the FCB Sequence Number that appears on the FCB that is assigned as a continuation FCB.

```
00000100   00 00 00 01  00 01 00 00  00 00 00 00  01 01 00 01   ................
00000110   52 4F 4F 54  44 49 52 00  00 44 49 52  00 01 00 00   ROOTDIR..DIR....
00000120   00 01 00 01  00 20 00 01  00 01 FF FF  07 BE 01 12   ..... .......>..
00000130   0B 20 37 3A  07 BE 01 12  0B 20 37 3A  00 00 00 00   . 7:.>... 7:....
00000140   00 00 04 00  00 00 04 00  00 00 00 01  00 00 01 2C   ...............,
00000150   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000160   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000170   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000180   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000190   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001A0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001B0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001C0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001D0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001E0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001F0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 DD D4   ..............]T
```

Usage ID Number

The number in this field indicates the following:

00 The FCB is available for assignment.

01 This is a main FCB, i.e., it is not a continuation FCB.

02   This is a continuation FCB.

```
00000100   00 00 00 01  00 01 00 00  00 00 00 00  01 01 00 01   ................
00000110   52 4F 4F 54  44 49 52 00  00 44 49 52  00 01 00 00   ROOTDIR..DIR....
00000120   00 01 00 01  00 20 00 01  00 01 FF FF  07 BE 01 12   ..... .......>..
00000130   0B 20 37 3A  07 BE 01 12  0B 20 37 3A  00 00 00 00   . 7:.>... 7:....
00000140   00 00 04 00  00 00 04 00  00 00 00 01  00 00 01 2C   ...............,
00000150   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000160   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000170   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000180   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000190   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001A0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001B0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001C0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001D0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001E0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001F0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 DD D4   ..............]T
```

Extent Count

This tells you the number of extents whose addresses appear on the FCB. In other words, this number pertains to the FCB, not the file. For example, if there are two FCBs for a particular file, the extent count on the main FCB would tell you only how many extent addresses appear on the main FCB. The Extent Count on the Continuation FCB would tell you the number of extent address appearing on the Continuation FCB.

```
00000100    00 00 00 01  00 01 00 00  00 00 00 00  01 01 00 01   ................
00000110    52 4F 4F 54  44 49 52 00  00 44 49 52  00 01 08 00   ROOTDIR..DIR....
00000120    00 01 00 01  00 20 00 01  00 01 FF FF  07 BE 01 12   ..... .......>..
00000130    0B 20 37 3A  07 BE 01 12  0B 20 37 3A  00 00 00 00   . 7:.>... 7:....
00000140    00 00 04 00  00 00 04 00  00 00 00 01  00 00 01 2C   ...............,
00000150    00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000160    00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000170    00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000180    00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000190    00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001A0    00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001B0    00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001C0    00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001D0    00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001E0    00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001F0    00 00 00 00  00 00 00 00  00 00 00 00  00 00 DD D4   ..............]T
```

File Type

This field tells you the kind of file to which the FCB is assigned:

00 00	Data file
00 01	Directory file
00 02	Image file
00 03	KSAM data file
00 04	KSAM key file
00 05	LL image file
00 06	Archive continuation file
00 07	[reserved]
00 08	System file
00 09	Archive file
00 0A - 00 FF	[reserved for the WMCS]
01 00 - FF FF	Available for user-defined file types

```
00000100   00 00 00 01  00 01 00 00  00 00 00 00  01 01 00 01   ................
00000110   52 4F 4F 54  44 49 52 00  00 44 49 52  00 01 00 00   ROOTDIR..DIR....
00000120   00 01 00 01  00 20 00 01  00 01 FF FF  07 BE 01 12   ..... .......>..
00000130   0B 20 37 3A  07 BE 01 12  0B 20 37 3A  00 00 00 00   . 7:.>... 7:....
00000140   00 00 04 00  00 00 04 00  00 00 00 01  00 00 01 2C   ...............,
00000150   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000160   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000170   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000180   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000190   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001A0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001B0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001C0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001D0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001E0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001F0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 DD D4   ..............]T
```

Filename

This field contains a hexadecimal representation of the filename.

```
00000100   00 00 00 01  00 01 00 00  00 00 00 00  01 01 00 01   ................
00000110   52 4F 4F 54  44 49 52 00  00 44 49 52  00 01 00 00   ROOTDIR..DIR....
00000120   00 01 00 01  00 20 00 01  00 01 FF FF  07 BE 01 12   ..... .......>..
00000130   0B 20 37 3A  07 BE 01 12  0B 20 37 3A  00 00 00 00   . 7:.>... 7:....
00000140   00 00 04 00  00 00 04 00  00 00 00 01  00 00 01 2C   ...............,
00000150   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000160   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000170   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000180   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000190   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001A0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001B0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001C0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001D0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001E0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001F0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 DD D4   ..............]T
```

File Extension

This field contains a hexadecimal representation of the file extension.

```
00000100   00 00 00 01  00 01 00 00  00 00 00 00  01 01 00 01   ................
00000110   52 4F 4F 54  44 49 52 00  00 44 49 52  00 01 00 00   ROOTDIR..DIR....
00000120   00 01 00 01  00 20 00 01  00 01 FF FF  07 BE 01 12   ..... .......>..
00000130   0B 20 37 3A  07 BE 01 12  0B 20 37 3A  00 00 00 00   . 7:.>... 7:....
00000140   00 00 04 00  00 00 04 00  00 00 00 01  00 00 01 2C   ...............,
00000150   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000160   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000170   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000180   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000190   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001A0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001B0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001C0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001D0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001E0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001F0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 DD D4   ..............]T
```

Version Number

This field contains a hexadecimal representation of the version number.

```
00000100   00 00 00 01  00 01 00 00  00 00 00 00  01 01 00 01   ................
00000110   52 4F 4F 54  44 49 52 00  00 44 49 52  00 01 00 00   ROOTDIR..DIR....
00000120   00 01 00 01  00 20 00 01  00 01 FF FF  07 BE 01 12   ..... .......>..
00000130   0B 20 37 3A  07 BE 01 12  0B 20 37 3A  00 00 00 00   . 7:.>... 7:....
00000140   00 00 04 00  00 00 04 00  00 00 00 01  00 00 01 2C   ...............,
00000150   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000160   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000170   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000180   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000190   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001A0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001B0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001C0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001D0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001E0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001F0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 DD D4   ..............]T
```

Directory's FCB Number

This field contains the number of the FCB assigned to the directory file that contains the file to which this FCB is assigned.

```
00000100   00 00 00 01  00 01 00 00  00 00 00 00  01 01 00 01   ................
00000110   52 4F 4F 54  44 49 52 00  00 44 49 52  00 01 00 00   ROOTDIR..DIR....
00000120   00 01 00 01  00 20 00 01  00 01 FF FF  07 BE 01 12   ..... .......>..
00000130   0B 20 37 3A  07 BE 01 12  0B 20 37 3A  00 00 00 00   . 7:.>... 7:....
00000140   00 00 04 00  00 00 04 00  00 00 00 01  00 00 01 2C   ...............,
00000150   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000160   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000170   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000180   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000190   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001A0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001B0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001C0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001D0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001E0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001F0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 DD D4   ..............]T
```

Directory's FCB Sequence Number

This field contains the Sequence Number of the FCB assigned to the directory file that contains the file to which this FCB is assigned.

```
00000100   00 00 00 01  00 01 00 00  00 00 00 00  01 01 00 01   ................
00000110   52 4F 4F 54  44 49 52 00  00 44 49 52  00 01 00 00   ROOTDIR..DIR....
00000120   00 01 00 01  00 20 00 01  00 01 FF FF  07 BE 01 12   ..... .......>..
00000130   0B 20 37 3A  07 BE 01 12  0B 20 37 3A  00 00 00 00   . 7:.>... 7:....
00000140   00 00 04 00  00 00 04 00  00 00 00 01  00 00 01 2C   ...............,
00000150   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000160   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000170   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000180   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000190   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001A0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001B0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001C0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001D0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001E0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001F0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 DD D4   ..............]T
```

Record Length

This tells you the length of the records that constitute the file.

```
00000100   00 00 00 01  00 01 00 00  00 00 00 00  01 01 00 01   ................
00000110   52 4F 4F 54  44 49 52 00  00 44 49 52  00 01 00 00   ROOTDIR..DIR....
00000120   00 01 00 01  00 20 00 01  00 01 FF FF  07 BE 01 12   ..... .......>..
00000130   0B 20 37 3A  07 BE 01 12  0B 20 37 3A  00 00 00 00   . 7:.>... 7:....
00000140   00 00 04 00  00 00 04 00  00 00 00 01  00 00 01 2C   ...............,
00000150   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000160   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000170   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000180   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000190   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001A0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001B0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001C0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001D0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001E0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001F0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 DD D4   ..............]T
```

Owner ID

This is the identification number assigned to the user who created (and thus owns) the file. This number comes from the user's User Authorization Record in the UAF.DAT file (read the description of the USERPROF Command).

```
00000100   00 00 00 01  00 01 00 00  00 00 00 00  01 01 00 01   ................
00000110   52 4F 4F 54  44 49 52 00  00 44 49 52  00 01 00 00   ROOTDIR..DIR....
00000120   00 01 00 01  00 20 00 01  00 01 FF FF  07 BE 01 12   ..... .......>..
00000130   0B 20 37 3A  07 BE 01 12  0B 20 37 3A  00 00 00 00   . 7:.>... 7:....
00000140   00 00 04 00  00 00 04 00  00 00 00 01  00 00 01 2C   ...............,
00000150   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000160   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000170   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000180   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000190   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001A0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001B0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001C0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001D0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001E0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001F0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 DD D4   ..............]T
```

Group ID

This is the group identification number assigned to the user who created the file (the owner). This number also comes from the user's User Authorization Record in the UAF.DAT file.

```
00000100   00 00 00 01  00 01 00 00  00 00 00 00  01 01 00 01   ................
00000110   52 4F 4F 54  44 49 52 00  00 44 49 52  00 01 00 00   ROOTDIR..DIR....
00000120   00 01 00 01  00 20 00 01  00 01 FF FF  07 BE 01 12   ..... .......>..
00000130   0B 20 37 3A  07 BE 01 12  0B 20 37 3A  00 00 00 00   . 7:.>... 7:....
00000140   00 00 04 00  00 00 04 00  00 00 00 01  00 00 01 2C   ...............,
00000150   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000160   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000170   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000180   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000190   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001A0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001B0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001C0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001D0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001E0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001F0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 DD D4   ..............]T
```

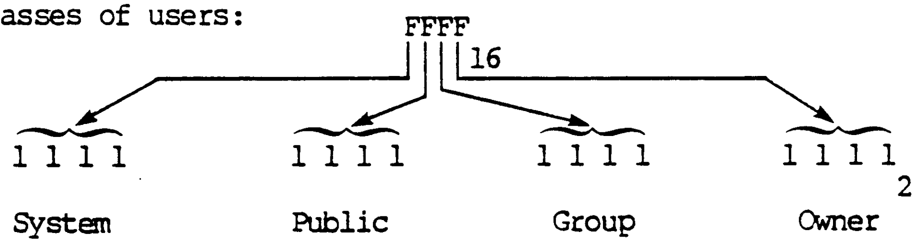
File Protection

This field tells you the privileges, regarding this file, that are assigned to each class of users.

These are the classes into which users on your system are divided:

Owner	The user who created the file.
Group	The group to which the owner is assigned.
Public	Users who are not members of the owner's group.
System	The system user(s).

Each byte in this field represents one of the foregoing classes of users:



For each class of users, there are four things that can be done with a file:

Delete	The file can be deleted.
Write	The contents of the file can be modified.
Read	The contents of the file can be perused.

Execute The program contained in the file (if it contains a program) can be executed.

Therefore, a single value in each hexadecimal digit in this field of the FCB tells you the privileges that that class of users has for the file. The following chart tells you how to interpret the values that can appear in each bit. In other words, one of the values in the lefthand column will appear in each of the four positions in this field of the FCB. The first value in the File Protection field of the FCB tells you the privileges the system user has, the second tells you the privileges the public has, etc. Thus, if an F appears in the first bit position. you know that the system user can delete, write, read, and execute the file (a one in any of the four columns next to the File Protection Value indicates that the privilege to perform the function represented by that column is granted). If an A appears for that first bit, the system can only delete and read the file.

File Protection Value	Delete	Write	Read	Execute
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
A	1	0	1	0
B	1	0	1	1
C	1	1	0	0
D	1	1	0	1
E	1	1	1	0
F	1	1	1	1

For example, F2 6F as the value appearing in this field of the FCB indicates that the system user has all privileges regarding the file, that the public can only read the file. the group can write and read the file, and the owner has all privileges.

```
00000100   00 00 00 01  00 01 00 00  00 00 00 00  01 01 00 01   ................
00000110   52 4F 4F 54  44 49 52 00  00 44 49 52  00 01 00 00   ROOTDIR..DIR....
00000120   00 01 00 01  00 20 00 01  00 01 FF FF  07 BE 01 12   ..... .......>..
00000130   0B 20 37 3A  07 BE 01 12  0B 20 37 3A  00 00 00 00   . 7:.>... 7:....
00000140   00 00 04 00  00 00 04 00  00 00 00 01  00 00 01 2C   ...............,
00000150   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000160   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000170   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000180   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000190   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001A0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001B0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001C0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001D0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001E0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001F0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 DD D4   ..............]T
```

File's Creation Date

This is the date the file was created. The date is given in hexadecimal values, according to the system clock time when the file was created.

```
00000100   00 00 00 01  00 01 00 00  00 00 00 00  01 01 00 01   ................
00000110   52 4F 4F 54  44 49 52 00  00 44 49 52  00 01 00 00   ROOTDIR..DIR....
00000120   00 01 00 01  00 20 00 01  00 01 FF FF  07 BE 01 12   ..... .......>..
00000130   0B 20 37 3A  07 BE 01 12  0B 20 37 3A  00 00 00 00   . 7:.>... 7:....
00000140   00 00 04 00  00 00 04 00  00 00 00 01  00 00 01 2C   ...............,
00000150   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000160   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000170   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000180   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000190   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001A0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001B0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001C0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001D0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001E0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001F0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 DD D4   ..............]T
```

File's Last Modification Date

This is the date, presented in the same manner as the creation date, on which the file was last modified.

```
00000100   00 00 00 01  00 01 00 00  00 00 00 00  01 01 00 01   ................
00000110   52 4F 4F 54  44 49 52 00  00 44 49 52  00 01 00 00   ROOTDIR..DIR....
00000120   00 01 00 01  00 20 00 01  00 01 FF FF  07 BE 01 12   ..... .......>..
00000130   0B 20 37 3A  07 BE 01 12  0B 20 37 3A  00 00 00 00   . 7:.>... 7:....
00000140   00 00 04 00  00 00 04 00  00 00 00 01  00 00 01 2C   ...............,
00000150   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000160   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000170   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000180   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000190   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001A0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001B0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001C0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001D0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001E0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001F0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 DD D4   ..............]T
```

Reserved

This field of the FCB is reserved for enhancement of the WMCS.

```
00000100   00 00 00 01  00 01 00 00  00 00 00 00  01 01 00 01   ................
00000110   52 4F 4F 54  44 49 52 00  00 44 49 52  00 01 00 00   ROOTDIR..DIR....
00000120   00 01 00 01  00 20 00 01  00 01 FF FF  07 BE 01 12   ..... .......>..
00000130   0B 20 37 3A  07 BE 01 12  0B 20 37 3A  00 00 00 00   . 7:.>... 7:....
00000140   00 00 04 00  00 00 04 00  00 00 00 01  00 00 01 2C   ...............,
00000150   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000160   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000170   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000180   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000190   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001A0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001B0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001C0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001D0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001E0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001F0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 DD D4   ..............]T
```

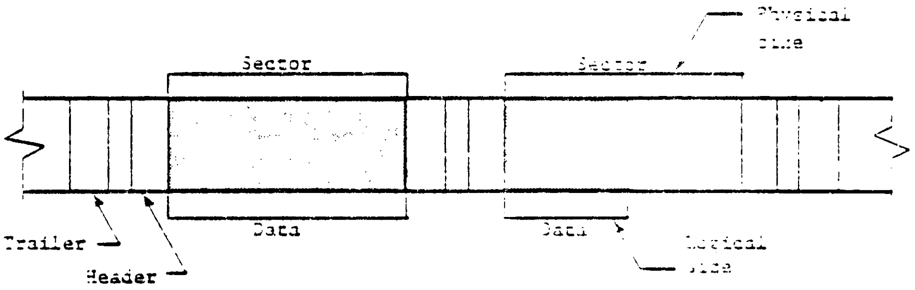
Physical Size

This tells you the actual, i.e., the physical, size of the file assigned to the FCB. In other words, if there are two FCBs for the file, the Physical Size field on the first, or primary, FCB tells you the physical size of the entire file.

The statement, in a directory listing, concerning file size is based on this number. However, what appears in the directory listing has been rounded to the nearest .1 kilobyte.

The physical size of a file is the number of bytes that are allocated to it. If a file does not fill a sector, that sector is still allocated to the file. Thus the physical end of the file, and the logical end of the file are not necessarily the same.

For example, the following illustration of a disk track indicates that while the data constituting the file actually occupy only 2.5 sectors, 3 sectors are allocated to the file.



```
00000100   00 00 00 01  00 01 00 00  00 00 00 00  01 01 00 01   ................
00000110   52 4F 4F 54  44 49 52 00  00 44 49 52  00 01 00 00   ROOTDIR..DIR....
00000120   00 01 00 01  00 20 00 01  00 01 FF FF  07 BE 01 12   ..... .......>..
00000130   0B 20 37 3A  07 BE 01 12  0B 20 37 3A  00 00 00 00   . 7:.>... 7:....
00000140   00 00 04 00  00 00 04 00  00 00 00 01  00 00 01 2C   ...............,
00000150   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000160   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000170   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000180   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000190   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001A0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001B0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001C0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001D0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001E0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001F0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 DD D4   ..............]T
```

Logical Size

This tells you the number of bytes in the file.

```
00000100   00 00 00 01  00 01 00 00  00 00 00 00  01 01 00 01   ................
00000110   52 4F 4F 54  44 49 52 00  00 44 49 52  00 01 00 00   ROOTDIR..DIR....
00000120   00 01 00 01  00 20 00 01  00 01 FF FF  07 BE 01 12   ..... .......>..
00000130   0B 20 37 3A  07 BE 01 12  0B 20 37 3A  00 00 00 00   . 7:.>... 7:....
00000140   00 00 04 00  00 00 04 00  00 00 00 01  00 00 01 2C   ...............,
00000150   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000160   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000170   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000180   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000190   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001A0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001B0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001C0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001D0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001E0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001F0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 DD D4   ..............]T
```

File Identification

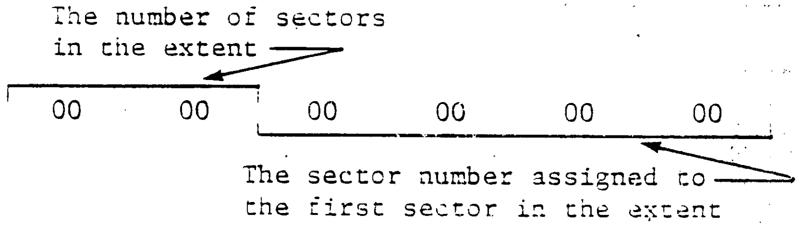
This field is reserved so that you can (if you wish) assign a value to it for the purpose of identifying the file.

```
00000100   00 00 00 01  00 01 00 00  00 00 00 00  01 01 00 01   ................
00000110   52 4F 4F 54  44 49 52 00  00 44 49 52  00 01 00 00   ROOTDIR..DIR....
00000120   00 01 00 01  00 20 00 01  00 01 FF FF  07 BE 01 12   ..... .......>..
00000130   0B 20 37 3A  07 BE 01 12  0B 20 37 3A  00 00 00 00   . 7:.>... 7:....
00000140   00 00 04 00  00 00 04 00  00 00 00 01  00 00 01 2C   ...............,
00000150   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000160   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000170   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000180   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000190   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001A0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001B0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001C0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001D0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001E0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001F0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 DD D4   ..............]T
```

Extents

An extent is a group of contiguous disk sectors belonging to the same file. In other words, the sectors in which a file is located are not necessarily contiguous. An extent is a group of contiguous sectors.

This field consists of 30. 6-byte subfields. Each subfield is broken down according to the following format:



The first two bytes in this field tell you how many sectors constitute the extent. The last four bytes give you the number, or address, of the first sector in that extent.

```
00000100   00 00 00 01  00 01 00 00  00 00 00 00  01 01 00 01   ................
00000110   52 4F 4F 54  44 49 52 00  00 44 49 52  00 01 00 00   ROOTDIR..DIR....
00000120   00 01 00 01  00 20 00 01  00 01 FF FF  07 BE 01 12   ..... .......>..
00000130   0B 20 37 3A  07 BE 01 12  0B 20 37 3A  00 00 00 00   . 7:.>... 7:....
00000140   00 00 04 00  00 00 04 00  00 00 00 01  00 00 01 2C   ...............,
00000150   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000160   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000170   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000180   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000190   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001A0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001B0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001C0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001D0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001E0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001F0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 DD D4   ..............]T
```

FCB Checksum

This is a notted checksum. i.e., the one's complement of the sum of the bytes in the FCB.

9.5.2 ROOTDIR DIR

/ROOTDIR/ is the root directory on your system. ROOTDIR.DIR is the directory file that contains a list of all the files in /ROOTDIR/.

Were you to initialize a diskette and then display the contents of ROOTDIR-DIR on your terminal screen, this is what would appear on your screen:

```
00000000  00 00 00 01  00 01 00 00  00 00 00 00  00 00 00 00   ................
00000010  00 00 2D 00  00 00 00 00  00 00 00 44  49 52 00 01   ..-........DIR..
00000020  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000030  00 00 46 43  42 00 00 00  00 00 00 53  59 53 00 01   ..FCB......SYS..
00000040  00 00 00 01  00 01 00 00  00 00 00 00  00 00 00 00   ................
00000050  00 00 52 4F  4F 54 44 49  52 00 00 44  49 52 00 01   ..ROOTDIR..DIR..
00000060  00 00 00 02  00 02 00 00  00 00 00 00  00 00 00 00   ................
00000070  00 00 46 43  42 42 49 54  4D 41 50 53  59 53 00 01   ..FCBBITMAPSYS..
00000080  00 00 00 03  00 03 00 00  00 00 00 00  00 00 00 00   ................
00000090  00 00 42 49  54 4D 41 50  00 00 00 53  59 53 00 01   ..BITMAP...SYS..
000000A0  FF FF FF FF  00 00 00 00  00 00 00 00  00 00 00 00   ................
000000B0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000000C0  FF FF FF FF  00 00 00 00  00 00 00 00  00 00 00 00   ................
000000D0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000000E0  FF FF FF FF  00 00 00 00  00 00 00 00  00 00 00 00   ................
000000F0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................


00000100  FF FF FF FF  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000110  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000120  FF FF FF FF  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000130  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000140  FF FF FF FF  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000150  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000160  FF FF FF FF  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000170  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000180  FF FF FF FF  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000190  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001A0  FF FF FF FF  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001B0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001C0  FF FF FF FF  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001D0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001E0  FF FF FF FF  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001F0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................


00000200  FF FF FF FF  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000210  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000220  FF FF FF FF  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000230  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000240  FF FF FF FF  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000250  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000260  FF FF FF FF  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000270  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000280  FF FF FF FF  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000290  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000002A0  FF FF FF FF  00 00 00 00  00 00 00 00  00 00 00 00   ................
000002B0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000002C0  FF FF FF FF  00 00 00 00  00 00 00 00  00 00 00 00   ................
000002D0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000002E0  FF FF FF FF  00 00 00 00  00 00 00 00  00 00 00 00   ................
000002F0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
```

```
00000300  FF FF FF FF  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000310  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000320  FF FF FF FF  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000330  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000340  FF FF FF FF  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000350  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000360  FF FF FF FF  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000370  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000380  FF FF FF FF  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000390  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000003A0  FF FF FF FF  00 00 00 00  00 00 00 00  00 00 00 00   ................
000003B0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000003C0  FF FF FF FF  00 00 00 00  00 00 00 00  00 00 00 00   ................
000003D0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000003E0  FF FF FF FF  00 00 00 00  00 00 00 00  00 00 00 00   ................
000003F0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
```

The following material describes each field in the display of a directory file's contents.

```
00000000  00 00 00 01  00 01 00 00  00 00 00 00  00 00 00 00   ................
00000010  00 00 2D 00  00 00 00 00  00 00 00 44  49 52 00 01   ..-........DIR..
00000020  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000030  00 00 46 43  42 00 00 00  00 00 00 53  59 53 00 01   ..FCB......SYS..
00000040  00 00 00 01  00 01 00 00  00 00 00 00  00 00 00 00   ................
00000050  00 00 52 4F  4F 54 44 49  52 00 00 44  49 52 00 01   ..ROOTDIR..DIR..
00000060  00 00 00 02  00 02 00 00  00 00 00 00  00 00 00 00   ................
00000070  00 00 46 43  42 42 49 54  4D 41 50 53  59 53 00 01   ..FCBBITMAPSYS..
00000080  00 00 00 03  00 03 00 00  00 00 00 00  00 00 00 00   ................
00000090  00 00 42 49  54 4D 41 50  00 00 00 53  59 53 00 01   ...BITMAP...SYS..
000000A0  FF FF FF FF  00 00 00 00  00 00 00 00  00 00 00 00   ................
000000B0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000000C0  FF FF FF FF  00 00 00 00  00 00 00 00  00 00 00 00   ................
000000D0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000000E0  FF FF FF FF  00 00 00 00  00 00 00 00  00 00 00 00   ................
000000F0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
```

Line numbers

Each line in the display is assigned a number that indicates the relative position (within the file) of the first byte on each line.

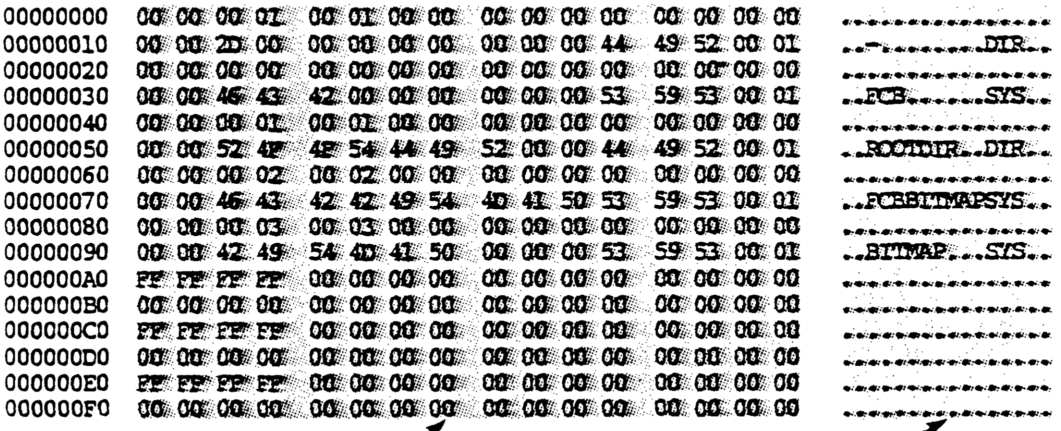
Record

Each record in a directory file consists of two lines, or 32 bytes.

Note that inasmuch as FCB.SYS, ROOTDIR.DIR, FCBBITMAP.SYS, and BITMAP.SYS are the first files created on any disk, these are always the first four files in ROOTDIR.DIR. Therefore. the first eight lines in ROOTDIR.DIR are allocated to these files.

Furthermore, the first record in a directory file is set aside for relative addressing, i.e., it designates the directory represented by the dash. -, used in relative addressing.

The last six lines in the foregoing diaplay are extra records created when the volume was initialized. Each newly created directory file contains several extra records. Records are added automatically when the directory file does not already contain enough records to accomodate new files.



```
00000000  00 00 00 01  00 01 00 00  00 00 00 00  00 00 00 00   ................
00000010  00 00 2D 00  00 00 00 00  00 00 00 44  49 52 00 01   ..-........DIR..
00000020  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000030  00 00 46 43  42 00 00 00  00 00 00 53  59 53 00 01   ..FCB......SYS..
00000040  00 00 00 01  00 01 00 00  00 00 00 00  00 00 00 00   ................
00000050  00 00 52 4F  4F 54 44 49  52 00 00 44  49 52 00 01   ..ROOTDIR..DIR..
00000060  00 00 00 02  00 02 00 00  00 00 00 00  00 00 00 00   ................
00000070  00 00 46 43  42 42 49 54  4D 41 50 53  59 53 00 01   ..FCBBITMAPSYS..
00000080  00 00 00 03  00 03 00 00  00 00 00 00  00 00 00 00   ................
00000090  00 00 42 49  54 4D 41 50  00 00 00 53  59 53 00 01   ..BITMAP...SYS..
000000A0  FF FF FF FF  00 00 00 00  00 00 00 00  00 00 00 00   ................
000000B0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000000C0  FF FF FF FF  00 00 00 00  00 00 00 00  00 00 00 00   ................
000000D0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000000E0  FF FF FF FF  00 00 00 00  00 00 00 00  00 00 00 00   ................
000000F0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
```

Bit field

This field contains the hexadecimal equivalents for the characters in the ASCII field.

ASCII field

Each character in this field is the ASCII equivalent of a hexadecimal value in the bit field.

```
00000000  00 00 00 01  00 01 00 00  00 00 00 00  00 00 00 00   ................
00000010  00 00 2D 00  00 00 00 00  00 00 00 44  49 52 00 01   ..-........DIR..
00000020  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000030  00 00 46 43  42 00 00 00  00 00 00 53  59 53 00 01   ..FCB......SYS..
00000040  00 00 00 01  00 01 00 00  00 00 00 00  00 00 00 00   ................
00000050  00 00 52 4F  4F 54 44 49  52 00 00 44  49 52 00 01   ..ROOTDIR..DIR..
00000060  00 00 00 02  00 02 00 00  00 00 00 00  00 00 00 00   ................
00000070  00 00 46 43  42 42 49 54  4D 41 50 53  59 53 00 01   ..FCBBITMAPSYS..
00000080  00 00 00 03  00 03 00 00  00 00 00 00  00 00 00 00   ................
00000090  00 00 42 49  54 4D 41 50  00 00 00 53  59 53 00 01   ..BITMAP...SYS..
000000A0  FF FF FF FF  00 00 00 00  00 00 00 00  00 00 00 00   ................
000000B0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000000C0  FF FF FF FF  00 00 00 00  00 00 00 00  00 00 00 00   ................
000000D0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000000E0  FF FF FF FF  00 00 00 00  00 00 00 00  00 00 00 00   ................
000000F0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
```

File's FCB Number

This field contains the number of the FCB in FCB.SYS that is assigned to the file whose name appears on the second line of the record in the ASCII field.

When FF FF FF FF, the hexadecimal value for -1, appears in this field, the record is unassigned, regardless of what may appear in the record's other fields.

```
00000000  00 00 00 01  00 01 00 00  00 00 00 00  00 00 00 00   ................
00000010  00 00 2D 00  00 00 00 00  00 00 00 44  49 52 00 01   ..-........DIR..
00000020  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000030  00 00 46 43  42 00 00 00  00 00 00 53  59 53 00 01   ..FCB......SYS..
00000040  00 00 00 01  00 01 00 00  00 00 00 00  00 00 00 00   ................
00000050  00 00 52 4F  4F 54 44 49  52 00 00 44  49 52 00 01   ..ROOTDIR..DIR..
00000060  00 00 00 02  00 02 00 00  00 00 00 00  00 00 00 00   ................
00000070  00 00 46 43  42 42 49 54  4D 41 50 53  59 53 00 01   ..FCBBITMAPSYS..
00000080  00 00 00 03  00 03 00 00  00 00 00 00  00 00 00 00   ................
00000090  00 00 42 49  54 4D 41 50  00 00 00 53  59 53 00 01   ..BITMAP...SYS..
000000A0  FF FF FF FF  00 00 00 00  00 00 00 00  00 00 00 00   ................
000000B0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000000C0  FF FF FF FF  00 00 00 00  00 00 00 00  00 00 00 00   ................
000000D0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000000E0  FF FF FF FF  00 00 00 00  00 00 00 00  00 00 00 00   ................
000000F0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
```

File's FCB Sequence Number

This field contains the FCB Sequence Number of the FCB in FCB.SYS that is assigned to the file whose name appears on the second line of the record in the ASCII field.

```
00000000  00 00 00 01  00 01 00 00  00 00 00 00  00 00 00 00   ................
00000010  00 00 2D 00  00 00 00 00  00 00 00 44  49 52 00 01   ..-........DIR..
00000020  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000030  00 00 46 43  42 00 00 00  00 00 00 53  59 53 00 01   ..FCB......SYS..
00000040  00 00 00 01  00 01 00 00  00 00 00 00  00 00 00 00   ................
00000050  00 00 52 4F  4F 54 44 49  52 00 00 44  49 52 00 01   ..ROOTDIR..DIR..
00000060  00 00 00 02  00 02 00 00  00 00 00 00  00 00 00 00   ................
00000070  00 00 46 43  42 42 49 54  4D 41 50 53  59 53 00 01   ..FCBBITMAPSYS..
00000080  00 00 00 03  00 03 00 00  00 00 00 00  00 00 00 00   ................
00000090  00 00 42 49  54 4D 41 50  00 00 00 53  59 53 00 01   ..BITMAP...SYS..
000000A0  FF FF FF FF  00 00 00 00  00 00 00 00  00 00 00 00   ................
000000B0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000000C0  FF FF FF FF  00 00 00 00  00 00 00 00  00 00 00 00   ................
000000D0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000000E0  FF FF FF FF  00 00 00 00  00 00 00 00  00 00 00 00   ................
000000F0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
```

FCB Number for deleted file

This is the number of the FCB that was assigned to the file that was assigned to this record in the directory file before the file that is currently assigned to this record.

Note that when a file is deleted from a directory, the record assigned to that file is not deleted, but becomes available for assignment to the next file added to the directory. Therefore, this field of the record indicates the number of the FCB that was assigned to the file previously assigned to the record.

```
00000000  00 00 00 01  00 01 00 00  00 00 00 00  00 00 00 00   ................
00000010  00 00 2D 00  00 00 00 00  00 00 00 44  49 52 00 01   ..-........DIR..
00000020  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000030  00 00 46 43  42 00 00 00  00 00 00 53  59 53 00 01   ..FCB......SYS..
00000040  00 00 00 01  00 01 00 00  00 00 00 00  00 00 00 00   ................
00000050  00 00 52 4F  4F 54 44 49  52 00 00 44  49 52 00 01   ..ROOTDIR..DIR..
00000060  00 00 00 02  00 02 00 00  00 00 00 00  00 00 00 00   ................
00000070  00 00 46 43  42 42 49 54  4D 41 50 53  59 53 00 01   ..FCBBITMAPSYS..
00000080  00 00 00 03  00 03 00 00  00 00 00 00  00 00 00 00   ................
00000090  00 00 42 49  54 4D 41 50  00 00 00 53  59 53 00 01   ..BITMAP...SYS..
000000A0  FF FF FF FF  00 00 00 00  00 00 00 00  00 00 00 00   ................
000000B0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000000C0  FF FF FF FF  00 00 00 00  00 00 00 00  00 00 00 00   ................
000000D0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000000E0  FF FF FF FF  00 00 00 00  00 00 00 00  00 00 00 00   ................
000000F0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
```

File type

This field tells you the kind of file to which the record is assigned:

00 00	Data file
00 01	Directory file
00 02	Image file
00 03	KSAM data file
00 04	KSAM key file
00 05	LL image file
00 06	Archive continuation file
00 07	[reserved]
00 08	System file
00 09	Archive file
00 0A - 00 FF	[reserved for the WMCS]
01 00 - FF FF	Available for user-defined file types

```
00000000  00 00 00 01  00 01 00 00  00 00 00 00  00 00 00 00   ................
00000010  00 00 2D 00  00 00 00 00  00 00 00 44  49 52 00 01   ..-........DIR..
00000020  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000030  00 00 46 43  42 00 00 00  00 00 00 53  59 53 00 01   ..FCB......SYS..
00000040  00 00 00 01  00 01 00 00  00 00 00 00  00 00 00 00   ................
00000050  00 00 52 4F  4F 54 44 49  52 00 00 44  49 52 00 01   ..ROOTDIR..DIR..
00000060  00 00 00 02  00 02 00 00  00 00 00 00  00 00 00 00   ................
00000070  00 00 46 43  42 42 49 54  4D 41 50 53  59 53 00 01   ..FCBBITMAPSYS..
00000080  00 00 00 03  00 03 00 00  00 00 00 00  00 00 00 00   ................
00000090  00 00 42 49  54 4D 41 50  00 00 00 53  59 53 00 01   ..BITMAP...SYS..
000000A0  FF FF FF FF  00 00 00 00  00 00 00 00  00 00 00 00   ................
000000B0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000000C0  FF FF FF FF  00 00 00 00  00 00 00 00  00 00 00 00   ................
000000D0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000000E0  FF FF FF FF  00 00 00 00  00 00 00 00  00 00 00 00   ................
000000F0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
```

File Identification

This field is reserved so that you can (if you wish) assign a value to it for the purpose of identifying the file.

```
00000000  00 00 00 01  00 01 00 00  00 00 00 00  00 00 00 00   ................
00000010  00 00 2D 00  00 00 00 00  00 00 00 44  49 52 00 01   ..-........DIR..
00000020  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000030  00 00 46 43  42 00 00 00  00 00 00 53  59 53 00 01   ..FCB......SYS..
00000040  00 00 00 01  00 01 00 00  00 00 00 00  00 00 00 00   ................
00000050  00 00 52 4F  4F 54 44 49  52 00 00 44  49 52 00 01   ..ROOTDIR..DIR..
00000060  00 00 00 02  00 02 00 00  00 00 00 00  00 00 00 00   ................
00000070  00 00 46 43  42 42 49 54  4D 41 50 53  59 53 00 01   ..FCBBITMAPSYS..
00000080  00 00 00 03  00 03 00 00  00 00 00 00  00 00 00 00   ................
00000090  00 00 42 49  54 4D 41 50  00 00 00 53  59 53 00 01   ..BITMAP...SYS..
000000A0  FF FF FF FF  00 00 00 00  00 00 00 00  00 00 00 00   ................
000000B0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000000C0  FF FF FF FF  00 00 00 00  00 00 00 00  00 00 00 00   ................
000000D0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000000E0  FF FF FF FF  00 00 00 00  00 00 00 00  00 00 00 00   ................
000000F0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
```

Reserved

This field is resrved for enhancement of the WMCS.

```
00000000  00 00 00 01  00 01 00 00  00 00 00 00  00 00 00 00   ................
00000010  00 00 2D 00  00 00 00 00  00 00 00 44  49 52 00 01   ..-........DIR..
00000020  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000030  00 00 46 43  42 00 00 00  00 00 00 53  59 53 00 01   ..FCB......SYS..
00000040  00 00 00 01  00 01 00 00  00 00 00 00  00 00 00 00   ................
00000050  00 00 52 4F  4F 54 44 49  52 00 00 44  49 52 00 01   ..ROOTDIR..DIR..
00000060  00 00 00 02  00 02 00 00  00 00 00 00  00 00 00 00   ................
00000070  00 00 46 43  42 42 49 54  4D 41 50 53  59 53 00 01   ..FCBBITMAPSYS..
00000080  00 00 00 03  00 03 00 00  00 00 00 00  00 00 00 00   ................
00000090  00 00 42 49  54 4D 41 50  00 00 00 53  59 53 00 01   ..BITMAP...SYS..
000000A0  FF FF FF FF  00 00 00 00  00 00 00 00  00 00 00 00   ................
000000B0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000000C0  FF FF FF FF  00 00 00 00  00 00 00 00  00 00 00 00   ................
000000D0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000000E0  FF FF FF FF  00 00 00 00  00 00 00 00  00 00 00 00   ................
000000F0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
```

Filename

This field contains the name of the file currently assigned to the record.

```
00000000  00 00 00 01  00 01 00 00  00 00 00 00  00 00 00 00   ................
00000010  00 00 2D 00  00 00 00 00  00 00 00 44  49 52 00 01   ..-........DIR..
00000020  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000030  00 00 46 43  42 00 00 00  00 00 00 53  59 53 00 01   ..FCB......SYS..
00000040  00 00 00 01  00 01 00 00  00 00 00 00  00 00 00 00   ................
00000050  00 00 52 4F  4F 54 44 49  52 00 00 44  49 52 00 01   ..ROOTDIR..DIR..
00000060  00 00 00 02  00 02 00 00  00 00 00 00  00 00 00 00   ................
00000070  00 00 46 43  42 42 49 54  4D 41 50 53  59 53 00 01   ..FCBBITMAPSYS..
00000080  00 00 00 03  00 03 00 00  00 00 00 00  00 00 00 00   ................
00000090  00 00 42 49  54 4D 41 50  00 00 00 53  59 53 00 01   ..BITMAP...SYS..
000000A0  FF FF FF FF  00 00 00 00  00 00 00 00  00 00 00 00   ................
000000B0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000000C0  FF FF FF FF  00 00 00 00  00 00 00 00  00 00 00 00   ................
000000D0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000000E0  FF FF FF FF  00 00 00 00  00 00 00 00  00 00 00 00   ................
000000F0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
```

File extension

This field contains the file extension of the file currently assigned to the record.

```
00000000  00 00 00 01  00 01 00 00  00 00 00 00  00 00 00 00   ................
00000010  00 00 2D 00  00 00 00 00  00 00 00 44  49 52 00 01   ..-........DIR..
00000020  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000030  00 00 46 43  42 00 00 00  00 00 00 53  59 53 00 01   ..FCB......SYS..
00000040  00 00 00 01  00 01 00 00  00 00 00 00  00 00 00 00   ................
00000050  00 00 52 4F  4F 54 44 49  52 00 00 44  49 52 00 01   ..ROOTDIR..DIR..
00000060  00 00 00 02  00 02 00 00  00 00 00 00  00 00 00 00   ................
00000070  00 00 46 43  42 42 49 54  4D 41 50 53  59 53 00 01   ..FCBBITMAPSYS..
00000080  00 00 00 03  00 03 00 00  00 00 00 00  00 00 00 00   ................
00000090  00 00 42 49  54 4D 41 50  00 00 00 53  59 53 00 01   ..BITMAP...SYS..
000000A0  FF FF FF FF  00 00 00 00  00 00 00 00  00 00 00 00   ................
000000B0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000000C0  FF FF FF FF  00 00 00 00  00 00 00 00  00 00 00 00   ................
000000D0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000000E0  FF FF FF FF  00 00 00 00  00 00 00 00  00 00 00 00   ................
000000F0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
```

Version number

This field contains the version number of the file currently assigned to the record.

9.5.3 FCBBITMAP.SYS

FCBBITMAP.SYS is the system file (hence the .SYS file extension) that the WMCS uses as a map of the condition and the use being made of the File Control Blocks (FCBs) in FCB.SYS.

This is what the contents of FCBBITMAP.SYS would look like were you to display the contents of that file on your screen:

```
00000000  FF FF FF CF  FF FF FF FF  FF FF FD FF  FF FF FF FF   ................
00000010  FF FF FF FF  FF FF FF FF  FF FF FF FF  FF FF FF FF   ................
00000020  FF FF FF FF  FF FF FF FF  FF FF FF FF  FF FF FF FF   ................
00000030  FF FD FB FF  FF FF FF 8F  FF FF FF F3  FF FF FF FF   ................
00000040  FD F7 FF FF  FF EF FF F7  FF 9F FF F7  BD FF D6 24   ...............$
00000050  EF F6 7D FC  18 C3 99 89  7B 7B F5 FB  8F FF 77 E3   ..}.....{{....w.
00000060  FB 0F F6 8F  FB FF FF BF  85 EE 9F FF  FE 23 FF FF   .............#..
00000070  FF FF AD FA  14 00 00 00  08 00 01 08  9F D1 2B 5F   ..............+_
00000080  F2 7C 5E FF  CB F7 9A 81  32 2F CA 14  0A 30 02 01   .|^.....2/...0..
00000090  00 DB FF FF  FF FF FF FF  FF FF FF FF  FF FF FF FF   ................
000C00A0  FF FF FF FF  FF FF FF FF  FF FE 80 00  00 00 00 00   ................
000000B0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000000C0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000000D0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000000E0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000000F0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................


00000100  00 00 00 30  00 00 00 00  00 00 02 00  00 00 00 00   ...0............
00000110  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000120  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000130  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000140  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000150  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000160  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000170  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000180  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000190  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001A0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001B0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001C0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001D0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001E0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001F0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
```

The following material explains the contents of FCBBITMAP.SYS.

```
00000000  FF FF FF CF  FF FF FF FF  FF FF FD FF  FF FF FF FF   ................
00000010  FF FF FF FF  FF FF FF FF  FF FF FF FF  FF FF FF FF   ................
00000020  FF FF FF FF  FF FF FF FF  FF FF FF FF  FF FF FF FF   ................
00000030  FF FD FB FF  FF FF FF 8F  FF FF FF F3  FF FF FF FF   ................
00000040  FD F7 FF FF  FF EF FF F7  FF 9F FF F7  BD FF D6 24   ...............$
00000050  EF F6 7D FC  18 C3 99 89  7B 7B F5 FB  8F FF 77 E3   ..}.....{{....w.
00000060  FB 0F F6 8F  FB FF FF BF  85 EE 9F FF  FE 23 FF FF   .............#..
00000070  FF FF AD FA  14 00 00 00  08 00 01 08  9F D1 2B 5F   ..............+_
00000080  F2 7C 5E FF  CB F7 9A 81  32 2F CA 14  0A 30 02 01   .|^.....2/...0..
00000090  00 DB FF FF  FF FF FF FF  FF FF FF FF  FF FF FF FF   ................
000000A0  FF FF FF FF  FF FF FF FF  FF FE 80 00  00 00 00 00   ................
000000B0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000000C0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000000D0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000000E0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000000F0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
```

```
00000100  00 00 00 30  00 00 00 00  00 00 02 00  00 00 00 00   ...0............
00000110  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000120  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000130  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000140  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000150  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000160  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000170  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000180  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000190  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001A0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001B0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001C0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001D0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001E0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001F0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
```

Line numbers

The line number indicates the relative position (within the file) of the first byte on the line. These numbers appear in the display for your convenience.

ASCII field

The characters in the ASCII field correspond to the byte values in the bit field.

Bit field

Read the bit field from left to right, beginning with the byte in the upper lefthand corner and moving down one line at a time.

Every byte in the first 16 lines of the foregoing display indicates the allocation status of eight FCBs.

For example, FF in the first byte in the field indicates the following allocation status for the FCBs indicated:

```
Byte        F                   F
       ┌────┴────┐         ┌────┴────┐
Bits   │ 1 1 1 1 │         │ 1 1 1 1 │
         | | | |             | | | |
FCBs     0 1 2 3             4 5 6 7
```

Note that when 1 is assigned to a bit, the FCB represented by the bit is allocated. A zero is assigned to a bit whose FCB is unallocated.

The CF in the fourth byte position in the foregoing display indicates the following allocation statuses for FCBs 25 - 32:

```
Byte          C                      F
         ┌────┴────┐           ┌────┴────┐
Bits     │ 1 1 0 0 │           │ 1 1 1 1 │
           / / \ \               / / \ \
FCBs     24 25 26 27           28 29 30 31
```

The second half of the record (the bottom 16 lines in the foregoing excerpt) is used to indicate whether or not FCBs are usable, i.e., good or bad. Each byte in this half of the record corresponds to the same FCBs represented by the bytes in the first 16 lines.

For example, the first byte on the first line in the record and the first byte on the 17th line in the record correspond to the same FCBs:

```
00000000  FF FF FF CF  FF FF FF FF  FF FF FD FF  FF FF FF FF   ................
00000010  FF FF FF FF  FF FF FF FF  FF FF FF FF  FF FF FF FF   ................
00000020  FF FF FF FF  FF FF FF FF  FF FF FF FF  FF FF FF FF   ................
00000030  FF FD FB FF  FF FF FF 8F  FF FF FF F3  FF FF FF FF   ................
00000040  FD F7 FF FF  FF EF FF F7  FF 9F FF F7  BD FF D6 24   ...............$
00000050  EF F6 7D FC  18 C3 99 89  7B 7B F5 FB  8F FF 77 E3   ..}.....{{....w.
00000060  FB 0F F6 8F  FB FF FF BF  85 EE 9F FF  FE 23 FF FF   .............#..
00000070  FF FF AD FA  14 00 00 00  08 00 01 08  9F D1 2B 5F   ..............+_
00000080  F2 7C 5E FF  CB F7 9A 81  32 2F CA 14  0A 30 02 01   .|^.....2/...0..
00000090  00 DB FF FF  FF FF FF FF  FF FF FF FF  FF FF FF FF   ................
000000A0  FF FF FF FF  FF FF FF FF  FF FE 80 00  00 00 00 00   ................
000000B0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000000C0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000000D0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000000E0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000000F0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
```

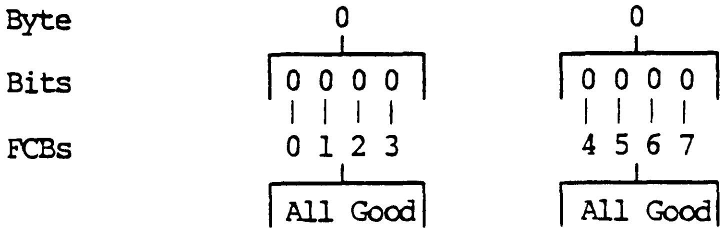
```
00000100  00 00 00 30  00 00 00 00  00 00 02 00  00 00 00 00   ...0............
00000110  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000120  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000130  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000140  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000150  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000160  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000170  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000180  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000190  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001A0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001B0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001C0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001D0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001E0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001F0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
```

Each byte in the second half of the bit field represents eight FCBs:

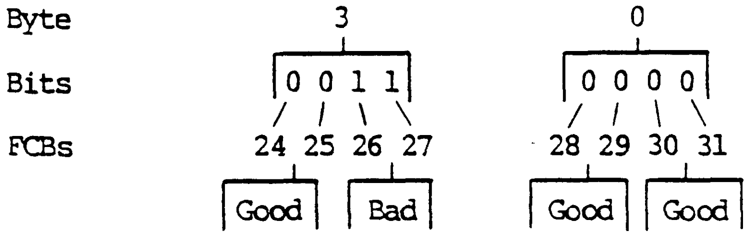
```
Bytes     0 0     0 0      0 0      3 0

FCBs     |0-7|  |8-15|  |16-23|  |24-31|
```

A zero is assigned to a bit to indicate that the FCB represented by the bit is usable; a one indicates that the FCB is bad, or unusable.

Therefore a 00 in the display indicates that the following values are assigned to the FCBs indicated:



The 30 in the fourth byte position in the foregoing display indicates the following:



9.5.4 BITMAP.SYS

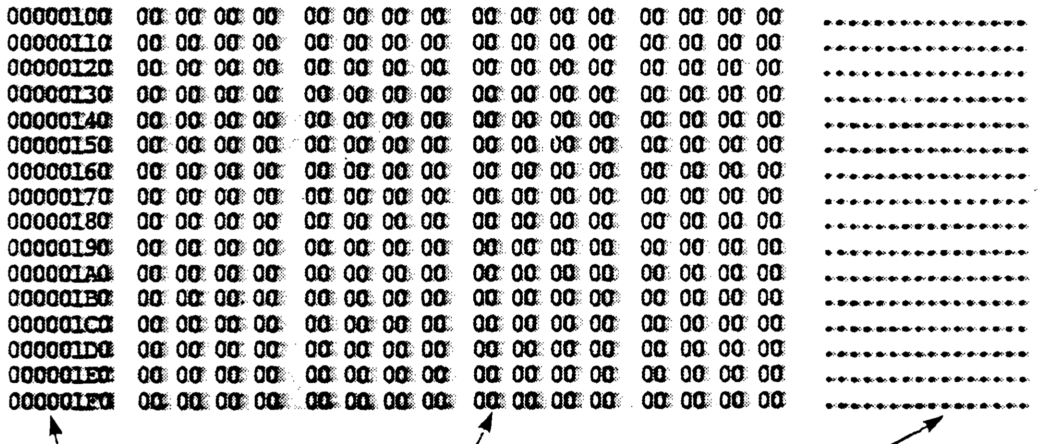
BITMAP.SYS is a system file (hence the .SYS file extension) that the WMCS uses as a map of the condition and the use being made of each sector on a disk.

This is what the contents of BITMAP.SYS look like when displayed on your terminal screen:

```
00000000  FF FF FF FF  FF BF F2 70  EF FF FF FF  FF FF FE FF   .......p........
00000010  FF FF FF FF  FF EF FF FF  FF FF FF FD  FB FF FF FF   ................
00000020  FF FF FF FF  FF BF FF F7  3F FF FF FF  FF FF FF FF   ........?.......
00000030  FF FF FF FF  FF FF 82 7F  FF FF FF FF  8F FF F9 FF   ...............
00000040  E3 FF FF FF  FF FF F0 03  FF FF FF FF  FF FF FF FF   ................
00000050  FF FF FF FC  7F FF FF FF  FF FF 00 7F  7F FF FF EF   ..........
00000060  FF FF FF F7  01 83 FF FF  8F FF FF FF  A3 FF FF FF   ................
00000070  FF FF FF FF  FC 1F FF CD  88 00 00 E0  7F 25 1F FF   ............%..
00000080  FE 7F FF F9  FF FC FF FF  F0 3F FF FF  FF F1 FF FF   .......?......
00000090  FF FF FF FF  FF FF FF DF  7F BF 97 FF  FF 7F FF FF   ............
000000A0  FF 3F FF 00  FF FF 7F 07  FF FF FF FF  FF FF FF FF   .?............
000000B0  F8 FE 00 03  FF FF FF FF  FF FF FF FF  FF FF FF FF   ................
000000C0  FF FF FF FF  FF FF FF FF  FF FF FF FF  FF FF FF FF   ................
000000D0  FF FF FF FF  FF FF FF FF  FF FF FF FF  F4 00 04 00   ................
000000E0  FF F8 18 1F  FF FF FF FF  FB 83 C6 83  FF E1 FF FF   ................
000000F0  FF FF FF FF  FF FF FF FF  FF FF FF FF  F0 00 01 FF   ................


00000100  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000110  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000120  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000130  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000140  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000150  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000160  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000170  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000180  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000190  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001A0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001B0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001C0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001D0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001E0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001F0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
```

The following material describes the contents of BITMAP.SYS.

```
00000000  FF FF FF FF  FF BF F2 70  EF FF FF FF  FF FF FE FF   .......p........
00000010  FF FF FF FF  FF EF FF FF  FF FF FF FD  FB FF FF FF   ................
00000020  FF FF FF FF  FF BF FF F7  3F FF FF FF  FF FF FF FF   ........?.......
00000030  FF FF FF FF  FF FF 82 7F  FF FF FF FF  8F FF F9 FF   ..............
00000040  E3 FF FF FF  FF FF F0 03  FF FF FF FF  FF FF FF FF   ................
00000050  FF FF FF FC  7F FF FF FF  FF FF 00 7F  7F FF FF EF   ..........
00000060  FF FF FF F7  01 83 FF FF  8F FF FF FF  A3 FF FF FF   ................
00000070  FF FF FF FF  FC 1F FF CD  88 00 00 E0  7F 25 1F FF   ...........%..
00000080  FE 7F FF F9  FF FC FF FF  F0 3F FF FF  FF F1 FF FF   .......?......
00000090  FF FF FF FF  FF FF FF DF  7F BF 97 FF  FF 7F FF FF   ............
000000A0  FF 3F FF 00  FF FF 7F 07  FF FF FF FF  FF FF FF FF   .?............
000000B0  F8 FE 00 03  FF FF FF FF  FF FF FF FF  FF FF FF FF   ................
000000C0  FF FF FF FF  FF FF FF FF  FF FF FF FF  FF FF FF FF   ................
000000D0  FF FF FF FF  FF FF FF FF  FF FF FF FF  F4 00 04 00   ................
000000E0  FF F8 18 1F  FF FF FF FF  FB 83 C6 83  FF E1 FF FF   ................
000000F0  FF FF FF FF  FF FF FF FF  FF FF FF FF  F0 00 01 FF   ................
```



```
00000100  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000110  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000120  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000130  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000140  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000150  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000160  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000170  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000180  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000190  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001A0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001B0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001C0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001D0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001E0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001F0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
```

Line numbers

The line number indicates the relative position (within the file) of the first byte on the line. These numbers appear in the display for your convenience.

ASCII field

The characters in the ASCII field correspond to the byte values in the bit field.

Bit field

Read the bit field from left to right. beginning with the byte in the upper lefthand corner and moving down one line at a time.

Every byte in the first 16 lines of the foregoing diplay indicates the allocation status of eiqht sectors.

ILLUSTRATION

Bytes	FF	FF	FF	FF	FF	FF	FF	FF
Sectors	0-7	8-15	16-23	24-31	32-39	40-47	48-55	56-63

For example, FF in the first byte in the field indicates the following allocation status for the sectors indicated:

Byte	F	F
Bits	1 1 1 1	1 1 1 1
Sectors	0 1 2 3	4 5 6 7

Note that when 1 is assigned to a bit, the sector represented by the bit is allocated. A zero is assigned to a bit whose sector is unallocated.

The BF in the sixth byte position on line one in the foregoing excerpt from a BITMAP.SYS file, indicates the following allocation statuses for sectors 41 - 49:

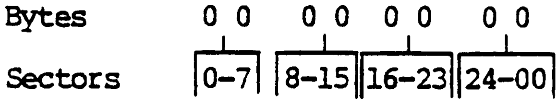
Byte	B	F
Bits	1 0 1 1	1 1 1 1
Sectors	40 41 42 43	44 45 46 47
Allocated?	Y N Y Y	Y Y Y Y

The second half of the record (the bottom 16 lines in the foregoing excerpt) indicates whether or not sectors are good. Each byte in this half of the record corresponds to the same sectors represented by the bytes in the first 16 lines.

For example, the first byte on the first line in the record and the first byte on the 17th line in the record correspond to the same sectors:

```
00000000  FF FF FF FF  FF BF F2 70  EF FF FF FF  FF FF FE FF   .......p........
00000010  FF FF FF FF  FF EF FF FF  FF FF FF FD  FB FF FF FF   ................
00000020  FF FF FF FF  FF BF FF F7  3F FF FF FF  FF FF FF FF   ........?.......
00000030  FF FF FF FF  FF FF 82 7F  FF FF FF FF  8F FF F9 FF   ..............
00000040  E3 FF FF FF  FF FF F0 03  FF FF FF FF  FF FF FF FF   ................
00000050  FF FF FF FC  7F FF FF FF  FF FF 00 7F  7F FF FF EF   ..........
00000060  FF FF FF F7  01 83 FF FF  8F FF FF FF  A3 FF FF FF   ................
00000070  FF FF FF FF  FC 1F FF CD  88 00 00 E0  7F 25 1F FF   ............%..
00000080  FE 7F FF F9  FF FC FF FF  F0 3F FF FF  FF F1 FF FF   .......?......
00000090  FF FF FF FF  FF FF FF DF  7F BF 97 FF  FF 7F FF FF   ............
000000A0  FF 3F FF 00  FF FF 7F 07  FF FF FF FF  FF FF FF FF   .?............
000000B0  F8 FE 00 03  FF FF FF FF  FF FF FF FF  FF FF FF FF   ................
000000C0  FF FF FF FF  FF FF FF FF  FF FF FF FF  FF FF FF FF   ................
000000D0  FF FF FF FF  FF FF FF FF  FF FF FF FF  F4 00 04 00   ................
000000E0  FF F8 18 1F  FF FF FF FF  FB 83 C6 83  FF E1 FF FF   ................
000000F0  FF FF FF FF  FF FF FF FF  FF FF FF FF  F0 00 01 FF   ................


00000100  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000110  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000120  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000130  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000140  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000150  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000160  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000170  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000180  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000190  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001A0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001B0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001C0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001D0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001E0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
000001F0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
```

ILLUSTRATION



A zero is assigned to a bit to indicate that the sector represented by the bit is good; a one indicates that the sector is bad.

Therefore, a 00 in the display indicates that the following values are assigned to the bits represented by the byte:

Byte	0	0
Bits	0 0 0 0	0 0 0 0
Sectors	0 1 2 3	4 5 6 7
	All Good	All Good

A 01 in the display would indicate the following for the bits it represents:

Byte	0	1	
Bits	0 0 0 0	0 0 0 1	
Sectors	0 1 2 3	4 5 6	7
	All Good	Good	Bad

9.6 HOW THE WMCS LOCATES DATA ON AN INITIALIZED DISK

When a user specifies a file designation as part of a request that the WMCS read data from a disk, the WMCS uses the disk's boot block to determine the sector number assigned to the first sector containing FCB.SYS.

That information is then given to the device driver assigned to the device in which the disk is mounted, the data from FCB.SYS are read into memory, and the WMCS finds the number of the FCB assigned to ROOTDIR.DIR.

Upon finding the FCB, the WMCS determines the sector number(s) assigned to ROOTDIR.DIR, and the data constituting ROOTDIR.DIR are read into memory. The WMCS then scans ROOTDIR.DIR to find a record containing a directory file whose name matches the name of the first directory file specified in the file designation as part of the original request.

Upon finding the record for that directory file, the WMCS reads the record to find the number of the FCB assigned to that directory file. The WMCS then gets the FCB (from FCB.SYS) for the directory file. Upon finding the FCB, the WMCS determines the sector number assigned to the first sector in the directory file. passes that information to the device driver, and the sectors containing the specified directory file are read into memory.

The WMCS then scans the contents of the directory file to find a record whose filename corresponds to that specified in the original request. Upon finding the record, the WMCS obtains the number of the FCB in FCB.SYS that is assigned to the file. The WMCS then gets the specified FCB (from FCB.SYS) reads the FCB, and opens the file.

CHAPTER 10

BACKUP

This chapter is written for inexperienced as well as experienced system managers.

10.1 INTRODUCTION

A periodic and systematic backup procedure is one of the most important responsibilities of the system manager.

This chapter contains a suggested schedule for creating backup copies of the files on your system and tells you how to file the tapes or diskettes containing the backup copies, the printouts of each backup, etc., so that backup copies are protected and easily accessible.

Read the descriptions of the BACKUP and RESTORE commands in the Multi-user Control System (WMCS) User Reference Manual for information on:

1. How to execute BACKUP and RESTORE.
2. How to create command files for daily, weekly, and monthly backups.

Appendix E in this manual contains sample command files to use for daily, weekly, and monthly backups.

10.2 THE SCHEDULE

The following suggested schedule ensures that you will never lose more than one day's work.

Your first task each working day should be to back up files on the system. Use the day of the week to determine what kind of backup to perform:

Monday: Weekly Backup

Includes all files that have changed since the previous weekly backup. See the description of the :SINCE= Switch for the BACKUP Command.

Tuesday - Friday: Daily Backups

Includes all files that have changed since the previous day.

First working day of the month: Monthly Backup

Includes all files on the system. The daily or weekly backup you would usually perform on this day is unnecessary.

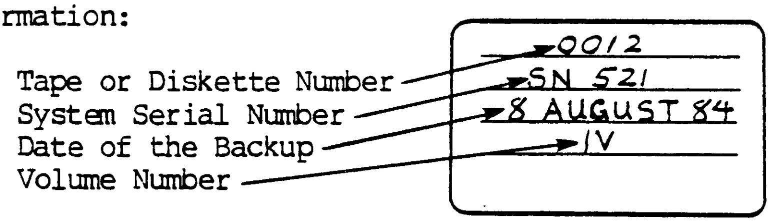
10.3 THE FILING SYSTEM

The following suggestions are easily adapted to tapes and diskettes:

1. Have enough tapes or diskettes to store to contain at least one month of weekly and daily backups (preferably two months), and at least three months of monthly backups (preferably 12 months).

   The number of tapes or diskettes you will need to store your system's files depends on the number and length of the files on the system.

2. Label each tape or diskette with the following information:



3. Always label Monday's backup, i.e., the weekly backup, as volume number 1. The daily backups for the rest of the week then become volumes 2, 3, 4, etc., respectively. The next weekly backup begins again with volume 1.

Volume 1 Monday's Weekly Backup for first week in July	Volume 2 Tuesday's Daily Backup	Volume 3 Wednesday's Daily Backup
Volume 4 Thursday's Daily Backup	Volume 5 Friday's Daily Backup	Volume 1 Monday's Weekly Backup for second week in July

The space required by these backups varies, e.g., Monday's weekly backup might be stored on volume 1 and part of volume 2, Tuesday's backup might fill the remainder of volume 2 and part of volume 3, Wednesday's backup might fill the remainder of volume 3, etc. (read the note below).

The first monthly backup tape or diskette should also be labeled beginning with volume one, because most monthly backups use space on more than one tape or diskette.

4. Always put weekly backups at the beginning of a tape or diskette.

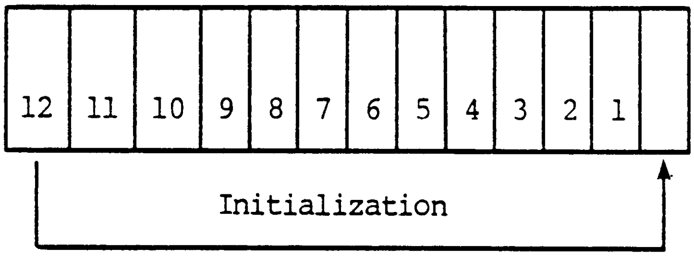
   You can then fill the remaining space on each tape or diskette with the daily backups or you may choose to begin each daily backup with a separate tape or diskette.

   NOTE: If you are using tapes as your backup storage medium, you may find it more practical to fill as much of the tape as possible by combining the daily backups (but always place the weekly backup at the beginning of a new tape).

   If you are using diskettes as your backup storage medium, you may decide to put each daily backup (as well as the weekly backup) at the beginning of a separate diskette.

5. Recycle tapes and diskettes.

   Initialize the oldest tape or diskette so that it becomes the newest backup volume.



NOTE: Always keep a minimum of one month's worth of backups.

6. Print the log file that is created each time you make a backup.

   Use the :LOG= Switch that is part of the BACKUP Command. You may find it helpful to name the log according to the kind of backup being performed, such as DAILY.LOG, WEEKLY.LOG, or MONTHLY.LOG.

   For example, to automatically print a backup log with a specified name of WEEKLY.LOG, place the following command in the backup command file:

   ```
   print weekly.log
   ```

   The cover sheet of the printout supplies the date and time of the backup as well as the name of the log file.

7. Write the tape or diskette numbers of all the backup volumes, recorded in the log, on the cover sheet of the printed log.

8. File or shelve the printout of the backup logs with the most recent log on top.

   When the oldest backup tape or diskette is reused, discard the printed backup log for that volume.

10.4 STORAGE

Correctly storing the backup tapes or diskettes is crucial to the success of a backup program. It does not matter how conscientious you are about making backups if the backups are lost or damaged through improper storage. Consider the following suggestions as you plan your backup storage:

1. Store monthly backups in a building separate from the one housing your computer system.

2. As long as your monthly backups are well protected, weekly and daily backups may be kept in a room separate from the computer system so that they are reasonably protected and still available for easy access.

3. Store the monthly, weekly, and daily printouts of the backup log file wherever it is most convenient for you to do so.

10.5 RESTORING LOST FILES

Backups become extremely important when a specific file or when all the files on a system need to be replaced. Follow the steps below to restore lost files.

10.5.1 Restoring A Single File

STEP 1: Beginning with the most recent backup log printout. locate the log that lists the latest version of the file to be replaced.

STEP 2: Mount the tape or diskette that corresponds to the log containing the file to be restored.

STEP 3: Use the RESTORE Command as described in the WICAT Multi-user Control System (WMCS) User Reference Manual.

10.5.2 Restoring The Whole Disk

STEP 1: Execute the RESTORE Command for the most recent monthly backup.

For example, if the most recent monthly backup was made the first working day in July, begin with it.

Use the Destination Parameter (read the description of the RESTORE Command) to restore all the files to their original directories.

STEP 2: Use the PU Command to delete any duplicates of restored files.

STEP 3: Execute RESTORE for the first weekly backup for the month.

Continuing with the example above, the first weekly backup made after the July monthly backup would probably be for the second week in July.

NOTE: The intervening daily backups, i.e., those made between the monthly backup and the first weekly backup, need not be restored.

STEP 4: Use the PU Command to delete duplicates of the restored files.

STEP 5: Execute RESTORE for the second weekly backup for the month, e.g., the backup made for the third week in July.

Once again, the intervening daily backups need not be restored.

STEP 6: Use PU to delete duplicates of the restored files.

STEP 7: Continue restoring any weekly backups up to and including the most recent weekly backup, and purqe after each backup is restored.

STEP 8: After you have restored the most recent weekly backup, restore the daily backups up to the current day. Purge after each backup is restored.

STEP 9: If you need files restored from earlier backups, restore them using the single file method described above.

1	2	3
Most Recent Monthly Backup	Weekly Backups up through most recent weekly backup	Daily Backups made after the last weekly backup

CHAPTER 11

TURNING OFF THE POWER TO THE COMPUTER

This chapter is written for both experienced and inexperienced system managers. It is appropriate to bring a system down and turn off the power if the machine will not be used for an extended period. The primary concern of the system manager, when determining what constitutes an extended period, is whether the stress on the system's hardware (when the power is turned on again) is justifiable. In other words, you should not turn the power off unless doing so is absolutely necessary, i.e., when boards are to be removed or added, when drive units are to be changed, etc.

Read the description of the SHUTDOWN Command in the WICAT Multi-user Control System (WMCS) User Reference Manual for information on how to prepare your system for turning the power off.

Chapter 12

How to Performance-Tune and Customize Your System

This chapter is written for experienced system managers.

Enhancing the Help Displays

A help display has been prepared for each CIP command and is included as part of the WMCS.

Each help display is stored in a file in /SYSHLP/ on the system disk. The filename for each file in /SYSHLP/ is the mnemonic for the command whose help display the file contains, e.g., DIR.HLP contains the help display for the DIR Command, DINIT.HLP contains the help display for the DINIT Command, etc. Note that .HLP is the file extension for each file in
/SYSHLP/.

Each file in /SYSHLP/ is a standard text file that can be edited by means of the VEW Program. Furthermore, when you develop a command, you may wish to prepare a help display for it and place the file containing the display in /SYSHLP.USERS/.

These are the objectives that governed the development of the help displays prepared by WICAT Systems (and which you may wish to consider in developing and emending help displays):

1. Prepare a help display for every standard command.

2. /SYSHLP/ should be reserved for standard WMCS help files because the files found in /SYSHLP/ are replaced with each release of WMCS. If you customize those files, they may be lost with the next release. User helps should be placed in /SYSHLP.USERS/.

3. The purpose of the help display is to refresh the user's memory, not to teach the user how to use the command. Thus, the help display does not replace the manual.

4. Each message is presented in the same format and consists of up to four items of information:

   a. A very brief description of the command.

   b. A "typical usage" example.

   c. Definitions of command parameters.

   d. Definitions of command switches.

If these objectives do not correspond to the needs of your operation, you can emend the help displays as you see fit. For example, if the users on your system prefer to have the displays in a language other than English, you may wish to translate the instructions into their own language.

Use the VEW Program to create the file that will contain the help display for a new command. Use the mnemonic for the command as the filename for the file, and .HLP as the file extension. For example, if the mnemonic for the new command is ADD, the text for the help display for that command should be stored in /SYSHLP.USERS/ADD.HLP. To call the text of that file to the terminal screen, the user types ADD? on the CIP command line.

Finally, use the VEW Program to add the mnemonic to the contents of
/SYSHLP/CMD.HLP and to add the mnemonic and a one-line description of the new command to the contents of /SYSHLP/HELP.HLP. This should be done each time you load a new version of the WMCS on your system.

Standard Command Files

There are three sets of system command files. Each set has a primary command file and some secondary command files referenced by the primary. Following is a summary of these system command files.

The three primary command files are named /SYSLIB/STARTUP.COM, /SYSLIB/LOGON.COM and /SYSLIB/LOGOFF.COM. These command files allow the system manager to configure the environment of his system. The command files marked with (W) should not be modified by the system manager. Those marked with (*) are maintained by the system manager. Those marked with (+) are the responsibility of each system user.

System Command Files

Executed at system boot		Executed at user logon		Executed at user log off	
STARTUP.COM	(W)	LOGON.COM	(W)	LOGOFF.COM	(W)
INSTALL.PRM	(W)	LOCALON.COM	(*)	LOCALOFF.COM	(*)
DEVICEUP.COM	(*)	USERUP.COM	(+)	USEROFF.COM	(+)
LOCALUP.COM	(*)				
APPLICUP.COM	(*)				

(W) - File maintained by WICAT. You should not modify these files
(*) - File maintained by System Manager.
(+) - File maintained by each system user.

Samples of system command files are listed in Appendix F of this manual.

Startup.com

This command file is executed each time the system boots. Its function is to prepare the system for use. You should not change this command file.

To prepare the system for use, STARTUP.COM references the following four command files:

INSTALL.PRM - This parameter file is referenced to install all of the system commands that need to be installed. YOU SHOULD NOT CHANGE THIS PARAMETER FILE.

DEVICEUP.COM - You will need to modify this command file to match the hardware configuration of your system. Its purpose is to assign system logical names associated with devices (e.g. sys$pipe and sys$print), to mount all devices that need to be mounted at boot time, and to set the device characteristics of all mounted devices. For instance, on a multi-user system, this command file would mount all of the terminals and set the terminal type on each of those terminals so that users could log on.

LOCALUP.COM - You will need to modify this command file to establish the environment you want for your system. The purpose of this command file is to assign system logical names that set up your system environment, and to execute any commands that your installation needs to have executed each time the system boots. For example, on systems without a battery backed clock, you should put in this command file a command to have the operator set the system time.

APPLICUP.COM - You will need to modify this command file to bring up the applications that you have on your system. For instance, if you have the Pascal compiler on your system, you should have a line in the APPLICUP.COM file that "brings up" the Pascal compiler, such as:

```
> @sys$disk/pascal/pascalup.com
```

This command file is automatically updated when you load an application onto your disk.

Logon.com

This command file is executed each time a user logs on to the system. Its function is to initialize the system environment for that user. You should not change this command file.

To initialize the system environment for a single user, LOGON.COM calls the following two command files:

LOCALON.COM - This command file executes any commands that need to be executed for every user. This is a common command file that is shared by all users in that it always gets executed when a user logs on. For instance, it can print a short news bulletin for each user to see when he logs on. You should change this command file to meet the needs of your system.

> NOTE: Make sure that this file and the LOCALOFF.COM file are protected against public write access. Because they are executed by all users (even those with sensitive privileges), a malicious modification of these two files could open serious flaws in system security.

USERUP.COM - This command file is found in the home directory of each authorized system user. It contains commands that that user would like to have executed when he logs on. Each user is responsible for setting up and changing his USERUP.COM file. The file named USERUP.COM in the /SYSLIB/ directory is for the system account. The system manager is responsible for its contents.

Logoff.com

This command file is executed each time a user logs off of the system. Its function is to perform any action necessary when users log off. You should not change this command file.

LOGOFF.COM calls the following two command files:

LOCALOFF.COM - This command file executes any commands that need to be executed for every user. This is a common command file that is shared by all users in that it always gets executed when a user logs off. You should change this command file to meet the needs of your system.

USEROFF.COM - This command file is found in the home directory of each authorized system user. It contains commands that that user would like to have executed when he logs off. Each user is responsible for setting up and changing his USEROFF.COM file. The file named USEROFF.COM in the /SYSLIB/ directory is for the system account. The system manager is responsible for its contents.

Process Scheduling

The following sections provide information to help you increase the system's process performance, i.e., shorten the amount of time a user waits for the system to respond to a request.

Process Priority

The priority assigned to a process determines how often that process is scheduled for execution by the processor. For example, processes at priority 0 are executed by the processor more frequently than processes at priority 1, etc.

Each process has two priorities, "base" priority and "current" priority. The priority assigned to a process when it is created is its base priority. When the process is running and becomes disk or I/O bound, its priority automatically "floats" up. This is the current priority of the process. If the process is disk bound, its priority floats up one level, e.g., from priority 7 to 6. If it is I/O or TTY bound, its priority floats up two, e.g., from priority 7 to 5. The current priority is used by the scheduler to choose the next process to execute. Thus, TTY-bound processes will be chosen before disk-bound processes for execution. Each process executes at its base priority.

In most cases, interactive processes, i.e., processes such as VEW functions or CIP commands that require the system and the user to alternately respond to one another, should be assigned the same base priority level so that their use of processor time is evenly distributed.

Critical tasks should be assigned a higher base priority, and background tasks such as compiles, backups, etc., should be assigned a lower base priority.

The following commands pertain to assigning process priority:

1. Use the PSTAT Command to determine the priority of a process.

2. Use the PSTAT Command to designate the base priority of a process.

3. Use the USERPROF Command to designate the base priority of the user process, i.e., to designate the priority that is assigned to the user process whenever the user logs on.

Scheduling Ratios

The scheduling ratio designates the number of times a process is allowed processor time for every time a lower level priority task is given processor time. There are sixteen possible scheduling priorities, and thus, fifteen possible ratios.

For example, if the ratio were 2,2,2,2,2,2,2,2,2,2,2,2,2,2,2, for every task at priority level 1, two tasks at level 0 are scheduled for processor time, and for every task at priority level 2, two tasks at level 1 are given processor time, etc.

If the ratio were 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1, all tasks would compete for processor time on an equal level.

The ratio 10,10,10,10,10,10,10,10,10,10,10,10,10,10,10, is the default ratio. The scheduling ratio can only be modified through the _prirat system call. Read the WMCS Programmer's Reference Manual for a detailed discussion of scheduling ratios.

Process Timeslice

The process timeslice is the maximum amount of time that a process is allowed to execute each time it is scheduled. The process is able to use its whole timeslice as long as it does not execute a system service call that causes the process to wait for another event to take place, such as a record to come in from the disk or a user to respond.

The timeslice is designated at log on time. Use the PSTAT Command to modify the timeslice for a process.

Assigning a larger timeslice does not affect the speed of execution for jobs that have to stop and wait for input or output before their timeslice is used, but does help jobs that can run and use all their allotted timeslice. When setting the timeslice, remember that the longer the timeslice the longer a user must wait for the computer to respond; the user's process must wait through longer timeslice periods before it is allowed processor time. The recommended timeslice range is 50 to 70 milliseconds.

Swapping

Swapping is a memory management technique designed to help systems which typically have large amounts of memory tied up by inactive processes. The SWAPPER utility allows more efficient use of system memory by moving inactive portions of memory to a disk file so that the memory may be utilized by active, executing processes. Memory swapping allows you to run more processes on a system with limited physical memory.

Without swapping, all processes must be resident in memory in order for any one of them to execute. This requires large amounts of memory, even though many of the processes are not executing. With swapping, only the executing processes must be resident in memory.

On a typical system, most of the processes are waiting for some signal or input before executing.

For example, the following PSTAT display shows most processes are in a "Waiting" or "ChildWait" state on this system:

```
Time:     28-Mar-1985 15:50:14.30                          Free memory: 340K
Up time:            0 06:58:54.32
  PID       Process Name          Port         Status     Size Prior Schedule Times
--------  ----------------  ---------------  ----------  ----  ----  -------  -----
00010001  Logflush          __LJUNEAU_TT0    Wait           4  2/2     62558   100
00010002  Que_Manager       __LJUNEAU_TT0    Wait          40  2/2       165   100
00010004  Swapper           __LJUNEAU_TT0    Wait          56  2/2         7   100
0001013C  SYS$CIP           __LJUNEAU_TT0    ChildWait     16  7/7       122    60
0001013B  Q1                __LJUNEAU_TT0    Execute      140  5/7       158    60
00010138  QMENU_SYSTEM      __LJUNEAU_TT0    ChildWait    124  7/7       312    60
00010137  @qoffice_SYSTEM   __LJUNEAU_TT0    ChildWait     16  7/7        62    60
0001002C  CIP_SYSTEM        __LJUNEAU_TT0    ChildWait     64  7/7      7465    60
```

Swapping enables your system to execute more programs than it has physical memory to execute, because it selects inactive processes which have a low probability of executing and "swaps" them out of memory to a disk file.

The following PSTAT display shows that several processes have been swapped:

```
Time:     28-Mar-1985 15:51:21.89                            Free memory: 84K
Up time:            0 07:00:01.91
  PID       Process Name          Port          Status    Size Prior Schedule Times
--------  ---------------    ---------------  ---------  ----  ----  -------  -----
00010001  Logflush           __LJUNEAU_TT0    Wait          4  2/2     62728    100
00010004  Swapper            __LJUNEAU_TT0    Wait         56  2/2        68    100
00010002  Que_Manager        __LJUNEAU_TT0    Wait         40  2/2       165    100
00010140  CIP_VEW            __LJUNEAU_TT0    Execute      16  7/7       113     60
0001013F  vew_SYSTEM         __LJUNEAU_TT0    ChildWait   460  7/7       239     60
0001013C  SYS$CIP            __LJUNEAU_TT0    ChildWait    16  7/7       177     60
0001013B  Q1                 __LJUNEAU_TT0    Swapped      64  7/7       158     60
00010138  QMENU_SYSTEM       __LJUNEAU_TT0    Swapped       4  7/7       312     60
00010137  @qoffice_SYSTEM    __LJUNEAU_TT0    Swapped       4  7/7        62     60
0001002C  CIP_SYSTEM         __LJUNEAU_TT0    Swapped      48  7/7      7465     60
```

These inactive processes remain on the disk in a file called SYS$DISK/ROOTDIR/SWAPFILE.SYS until they become executable. When SWAPPER brings a process from the disk back into memory to be executed, there is a slight delay in response time. However, in most cases this delay will not be a serious inconvenience to the user.

Swapping Order

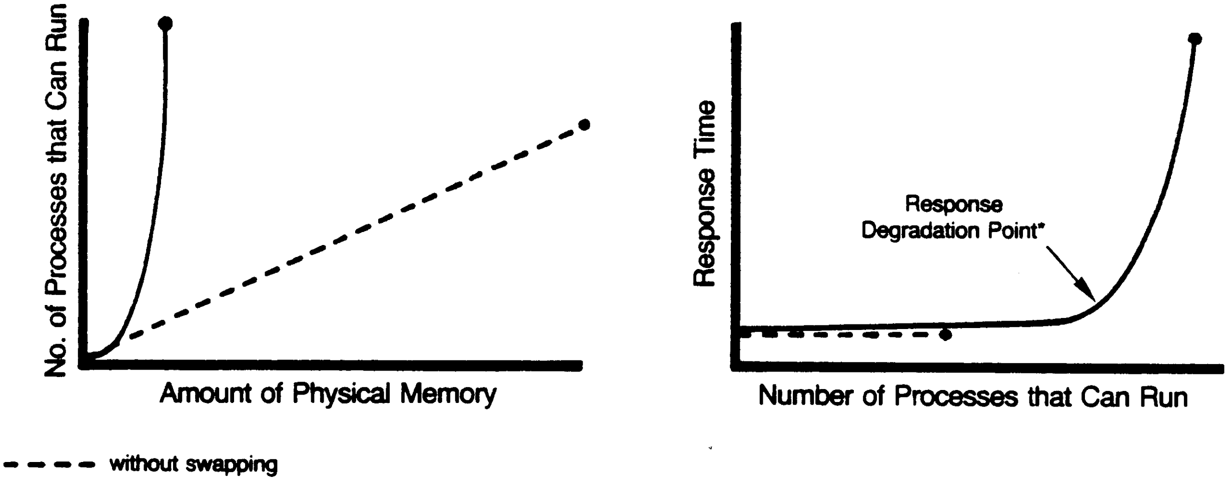
Swapping has been designed to impact the performance of a system as little as possible by intelligently selecting processes to be swapped. Processes are moved to the disk based upon a hierarchy of probabilities that they will not be needed in the immediate future. Process memory is only moved to the swap file when memory requests are made of the WMCS and sufficient system memory is unavailable to fill the requests. The order in which memory is moved to the swapfile is:

1. Memory belonging to a process in a childwait state.
2. Memory belonging to a process in a hibernate state.
3. Memory belonging to a process in an I/O wait state.
4. Memory belonging to a named shared memory region.
5. Memory belonging to an executable process.

Within each category, memory belonging to the process or named-shared-memory which has been in that category the longest is the first to be swapped. Only the amount of memory necessary to fill a request is actually swapped. This minimizes unnecessary movement of data. Private memory pages are swapped out before shared pages are considered.

Thus, more users can be supported on a system with swapping, but only at the expense of response time. Additional users' processes can be added with relatively little response degradation until the point is reached where more active processes exist than will fit in physical memory. From that point on, serious degradation will be noticeable as swapping becomes prerequisite to the execution of a timeslice for each process. This slowdown only occurs when many processes are running. The CPU is actually spending more time moving processes from disk to memory than in executing the processes.

The following charts illustrate the effects of swapping on system performance:



*Point at which memory is completely occupied by active processes

If most of the processes on your system are active most of the time, swapping would cause your system performance to suffer. CPU time is wasted if there are too many disk-to-memory swaps. Therefore, swapping is only effective on systems with many inactive processes that can be moved to the swapfile and remain there for a period of time.

Although swapping allows more processes to run in a given amount of physical memory, it does not allow any single process to use more memory than actually exists on the system. Before a process can begin execution, ALL of the pages of memory belonging to that process must be resident in physical memory. The sum of memory required by all processes and the WMCS cannot exceed the sum of physical memory available plus the size of the swapfile.

Swapping Immunity

Swapping is not desirable for every process. For example, if the SWAPPER process is moved to disk, it would never get swapped back into physical memory. Similarly, the nature of other processes may require that they be immune to swapping. The DM utility and processes that require a guaranteed immediate response should not be swapped. Swapping immunity can be arranged by setting the process attribute bits either at process creation time or as the process runs. The CIP allows you to create a non-swappable process on the command line. A PSTAT with appropriate privileges can change the attribute bits of a running process with the

:attribute=noswap switch. USERPROF can arrange for all processes belonging to a user to be non-swappable by default. See the documentation on these utilities in the WMCS User's Reference Manual for details.

With swapping there is the possibility of process deadlock. A deadlock can occur when many non-swappable processes are running at the same time. Once a process is moved to the swapfile, it cannot be killed until it is swapped back into memory. However, if non-swappable processes are using up all of the memory, no swapfile processes can be swapped back into memory to be killed.

For another example of process deadlock, suppose you have 1 Mbyte (1000 Kbytes) of memory on your system. Two swappable processes, A and B, are now running. Process A requires 800 Kbytes to execute, and process B requires 200 Kbytes. Both processes are using up all of the physical memory. Process C, a non-swappable process, requires 900 Kbytes of memory to run. SWAPPER moves processes A and B out to the swapfile to make room for process C. But process C needs input from process A, so process C goes into an I/O wait state because there is not room to swap process A back into memory.

The SWAPPER utility can be enabled and disabled at any time at the system manager's discretion. It can be started up at boot time by editing the DEVICEUP.COM file. The disk space used by the swapper is system manager configurable. See the command description for SWAPPER in the WMCS User's Reference Manual for more information.

Disk Performance

The following sections describe how to optimize disk performance on your system.

Cache Size

Cache size is the number of disk sectors that, for any given disk, are kept in memory. The size of the cache is determined when a disk is initialized with the DINIT Command, and can be changed using the BTUP Command (described in the WMCS User's Reference Manual.

The larger the cache size, the greater the likelihood that a requested sector is already in memory. If a requested sector is already in memory, you can obtain its data much faster than you could if you had to wait for the data to be retrieved from the disk.

Inasmuch as a larger cache uses more memory, you must decide whether the increase in speed is worth the loss of memory space. If you consistently have a large amount of unused memory, you may want to increase the size of the cache. Read the chapter (in this manual) on Initialization of Media for more information on cache size.

User Cache

The user cache is the maximum number of sectors from the total cache that can be used in a single operating system request made by a process. A user cache of 1 - 4 kilobytes (1 - 4 on a disk with 1024-byte sectors and 2 - 8 on a disk with 512-byte sectors) is recommended. Setting the user cache above 4 kilobytes provides no significant improvement in system performance. In fact, it may prove detrimental in that it increases the likelihood that frequently used sectors will not be available in the disk cache.

Autoflush Flag

The autoflush flag determines whether the sectors in the disk cache are written upon request or automatically after each critical operation. This is determined, for each disk, when the disk is initialized and can be modified with the BTUP Command.

Setting the autoflush flag parameter to the DINIT and BTUP commands to NO improves performance because updates to the disk are batched and written at one time instead of every time a file is created, closed, deleted, or renamed.

The trade off involves system integrity versus an increase in system speed. The LOGFLUSH command can be used to control the integrity of the file system inasmuch as the LOGFLUSH Program updates the disk every 30 seconds. You can also use the RECOVER Command following any event, such as a system crash, that might jeopardize the integrity of the file system.

Readahead

To improve performance for processes that read files sequentially, set the readahead parameter to the DINIT and BTUP commands to YES.

Ialloc and Alloc

The Ialloc (initial sector allocation) and Alloc (subsequent sector allocation) parameters to the DINIT and BTUP commands should typically be set for 20 to 50 sectors each. If a file does not use all of the sectors allocated to it, the unused portion is removed when the file is closed.

The advantages for having large Ialloc and Alloc values are:

1. It takes less time to allocate large numbers of sectors at one time than it does to allocate small numbers of sectors as they are needed. Therefore, system performance is improved.

2. When a large value is specified for IALLOC and ALLOC, there is less fragmentation of the files on the disk. The file system can take advantage of contiguous blocks of available sectors.

3. There is no wasted disk space:

   a. Extra sectors that would have been allocated to a file were the file extended are released when the file is closed.

   b. The file system does not require large groups of contiguous sectors.

The primary disadvantage to having large Ialloc and Alloc values is that if all the files on the system are small, system speed is decreased by the amount of time it takes to free the unused sectors.

Asynchronous Disk Read

When a file is read using normal file read requests, the disk sector containing the requested data is moved via direct memory access (DMA) from the disk into the disk cache physical memory. The data are then moved by the CPU from the disk cache physical memory to the user process logical memory. The file data are moved twice: once by DMA and once by the CPU.

To improve the performance of programs which read files sequentially, asynchronous file reads can be used. When a file is read asynchronously (so-called "fast read"), the disk sector is moved via DMA directly from the disk into the physical memory allocated to the user process logical memory. The file data are moved only once: by DMA. This is much faster than a normal read since the data are moved only once and the CPU did not have to move the data at all. It was able to do other things during the read.

The advantages of asynchronous reads are that one or more reads may be requested of the operating system and while the reads are being performed, the user process may do something else. When the process is at a point where it needs the requested data, it does a "fast read" wait (See the _FRDWAIT system call description in the WMCS Programmer's Reference Manual) which returns when DMA has transferred all of the data.

The disadvantages of asynchronous reads are that only disk sectors are transferred and the user buffer for receiving the data must be entirely contained within a 4-Kbyte page of physical (logical) memory. Therefore, the most data that can be read with an asynchronous read is 4 Kbytes.

Disk Usage

This section provides suggestions for efficiently monitoring and using disk space.

Three things typically account for a decrease in disk space:

1. The existence of multiple versions of a file.

2. The existence of extra copies of the same file in several areas on the disk.

3. The existence of obsolete and unused files.

Several solutions can be used to overcome insufficient disk space:

1. Buy more disks.

2. Delete unused files.

3. Transfer little-used files onto removable media such as diskettes, tapes, etc.

The following procedure may be helpful if you choose to free up disk space by deleting unused files and/or transferring seldom-used files to other media:

1. Use the DIR Command to find areas of extensive disk usage.

2. Print the directory listings and use the printout to locate large directories that use a great amount of space.

3. Send copies of the directory printouts to the users who own the directories.

4. Ask each user to examine their directories and:

    a. Delete unused files.

    b. Allow you to copy little-used files onto a removable volume.

5. Remind users to use the PU Command regularly to delete unneeded versions of files.

Networks

Networking can require and consume a lot of system resources; mainly the CPU and memory. There are many options available to the system manager for managing network performance on his system. All of the options are managed through the NSYSPROF utility.

Maximum Receive Cache Buffers

The network "maximum receive cache buffers" specifies the maximum receive buffers that will ever be allocated. When the first physical network device is mounted, the number of receive cache buffers specified by "minimum receive cache buffers" are allocated immediately. If the maximum receive cache buffers is greater than the minimum receive cache buffers and more buffers are needed during peak usage, more receive buffers are allocated (up to the maximum).

The size of a network receive buffer is 1500 bytes. Five buffers occupy 8 Kbytes, or two 4-Kbyte physical pages. When receive buffers are allocated, they are allocated five buffers at a time, i.e. two pages of memory at a time. Therefore, the difference between the minimum and maximum receive cache buffer sizes should be a multiple of five.

The advantage of a large number of receive buffers is that more virtual circuits may be in use at the same time.

The disadvantage of a large number of receive buffers is that more memory is consumed by the operating system and less memory is available for user programs.

If your system has a large amount of unused memory and requires that a lot of users on other systems access your system, specify a larger value for the "maximum receive cache buffers".

Default Receive Window Size

The "default receive window size" defines how many receive buffers will be used by each active circuit. If this value is set to one, each circuit can only use one buffer. Thus, if a data packet is written to a remote system, the remote process must read the packet before the local process can write another data packet.

The advantage of a larger receive window size is that a local or remote process can write a larger number of data packets before the receiving remote or local process must read them. This increases network throughput.

The disadvantages of a larger receive window size are that more memory is consumed per virtual circuit and that either more memory must be allocated to the maximum receive cache buffers or fewer virtual circuits can be active at the same time. Also, there is a point of diminishing returns with a larger receive window size, i.e. larger and larger numbers increase the throughput by smaller and smaller percentages.

Optimal throughput can be achieved by setting the receive window size to at least 3. A window size of 5 or greater is probably past the point of diminishing returns.

Virtual Circuits

The operating system uses virtual circuits for so-called "SVC communications", i.e., supervisor calls (SVCs) are communicated between machines via virtual circuits. The first and last SVC virtual circuit fields specify a range of virtual circuit numbers that may be used by WMCS. All other virtual circuits are reserved for users.

These two fields provide a way of putting an absolute limit on the number of virtual circuits that WMCS will use.

The advantage of a large number of virtual circuits reserved for SVC calls is that more users may simultaneously access resources on other systems at the same time.

The disadvantage of a large number of virtual circuits is that more memory is consumed (total buffers = maximum of "maximum receive cache buffers" and "default window size" times the number of active virtual circuits).

Local Remote Network ID

The remote network ID (RNID) is a system-manager defined number which can be used to group systems in a network together. Systems with the same RNID number form a logical group. Therefore, users of particular systems which access resources among themselves much more often than they access resources on all systems on the network should probably be grouped together, i.e., their systems should have the same RNID number.

Every time a new system comes up on the network, its nodename, site ID and RNID are exchanged with every system on the network. Therefore, every system's name, site ID and remote network ID are known to every other system without having to retrieve the information from each other every time the information is required.

Programmers developing applications can speed system searches, and therefore network throughput, by using the supervisor calls which deal with RNIDs. They are _RNIDLST and _RSIDLST. The _RNIDLST SVC returns all of the different RNIDs currently in use among the systems on the network. The _RSIDLST SVC returns the site IDs of all systems which have a particular RNID. Using these SVCs and the RNIDs, programs can be written so that they only deal with the systems of interest (a particular RNID) instead of searching every system on the network to see if it is one which the program is interested in. This can save a lot of needless overhead and increase apparent network and system throughput.

CHAPTER 13

RELEASE NOTICES AND PRODUCT INSTALLATION

13.1 AUDIENCE

The information in this chapter is for both inexperienced and experienced system managers.

13.2 NOTICES AND INSTALLATION

Every software product release or update you receive from WICAT Systems is accompanied by a release notice. A release notice has four purposes:

1. To document the differences between the previous release of the product and the new release.

2. To inform you of any bugs, i.e., software problems, in the last release that have been fixed in the new release. and to indicate the presence of any known bugs in the new release.

3. To provide instructions on how to install the new release on your system.

4. To provide instructions on how to create a backup copy of the new release.

When you receive a product release. carefully read the instructions **before** you attempt to install the new release.

CHAPTER 14

CONFIGURING THE WMCS

The WMCS is made up of modules, or files, that handle various classes of devices as well as various software components, e.g., the Keyed Sequential Access Method (KSAM), etc. Therefore, the devices and the software components that constitute your system determine what WMCS modules must be loaded when your system is booted in order for your system to operate. Configuration of the WMCS is a matter of selecting those modules that handle the kinds of devices and software options that you have on your system.

14.1 WHAT CONFIGURATION ACCOMPLISHES

These are the three aspects of system configuration:

1. Apprising the WMCS of the system's hardware components.

   These are the kinds of hardware variables that the WMCS must be aware of:

   a. The amount of memory.

   b. The kinds of controller boards.

   c. The number of TTY ports.

   d. The kind(s) of disk drives. e.g., 5- or 8-inch floppy-disk drives, 10-, 13-, 15-, 20-, 40- 80-, 160-, or even 421-Mb. drives (some systems have a single drive; others have several).

   e. The kind(s) of tape drive(s), if any, e.g., cartridge or cipher.

   f. Whether or not the system has a hardware floating point device.

g. Whether or not the system has the battery-powered calendar clock.

2. Loading appropriate portions of the WMCS.

This aspect of system configuration involves the WMCS modules that are loaded during the boot process. The kind of system you have determines which modules should be loaded. For example, one version of the KERNEL, a file called KERNELBUG, contains a built-in WMCS debugging capability.

NOTE: KERNELBUG is **not** for use in debugging user programs.

Furthermore, some systems need a tape class handler, some require the Keyed Sequential Access Method (KSAM) in addition to the standard random access method. Some systems require the original floating point emulation package that uses the 1010 trap instruction; other systems have been converted to the IEEE standard floating point.

3. Establishing the appropriate software environment.

Some systems are general purpose, while others are dedicated to a specific application. The dedicated system may be set up as a turnkey system, i.e., a system that, when turned on, begins immediately to execute the application to which the system is dedicated. For example, a system can be set up to automatically begin with the word processing program rather than requiring the user to execute the program.

Some users may wish to use the CIP as their standard user interface with WMCS; others may choose a different interface. Your system may require that users log in or it may not. It may or may not require passwords. Logical name assignments can even be specified so that the working environment of a system can be tailored to the needs and wishes of its users.

Finally, the programming languages and applications software can vary from system to system.

14.2 CONFIGURATION COMPONENTS

While most system configuration takes place automatically during the boot process. some aspects of system configuration can be altered while the system is running:

1. The SYSPROF Program allows you to modify these aspects of system configuration and device configuration:

    a. The operating system modules to be loaded when the system is booted.

    b. The system boot-device driver.

    c. The number of TTY ports to poll.

2. How to establish a turnkey system.

3. If necessary. modify the system command files to establish a customized working environment.

4. Use system utilities to mount and dismount devices while the system is running.

14.2.1 Automatic Configuration At Boot Time

The system manager has no responsibility for functions automatically performed by the system. For example, the system performs the following functions:

1. Locates all of the memory boards in the system, performs the setup functions that make the boards usable, and determines the board type and amount of memory.

2. Determines the number of TTY ports, and initializes each to a standard state, i.e., one stop bit, one start bit, disabled parity, and 9600 baud.

3. Detects the presence or absence of the floating point (hardware) board, and initializes the board if it is present.

4. Detects the presence or absence of the calendar clock and determines whether it is running.

14.2.2 The System Configuration File

Each system contains a system configuration file, /ROOTDIR/SYSCONFIG.nnn.1 (the nnn represents the kind of system you have, e.g., 156 for a System 150-6WS).

This file defines which operating system modules are loaded when the system is booted, the name of the device driver for the boot device, and the current year. SYSCONFIG.nnn may be inspected and modified with the SYSPROF Command.

Note that the system uses only verion 1 of this file. Confusion may result if you inadvertently create multiple copies of this file.

14.2.3 The Device Configuration File

Each system contains a device configuration file. /ROOTDIR/DEVCONFIG.nnn.1 (the nnn represents the kind of system you have). This file defines the standard devices on your system. It contains a record for each device. Each record contains:

1. The devicename. This name may contain up to nine alphanumeric characters. Spaces are not allowed. The following is a list of recommended devicenames:

5 1/4 inch floppies	_dx0, _dx1, _dx2, ...
8 inch floppies	_df0, _df1, _df2, ...
5 1/4 inch winchesters	_dc0, _dc1, _dc2, ...
SMD disk drives	_ds0, _ds1, _ds2, ...
All serial ports	_tt0, _tt1, _tt2, ...
1/2 inch magnetic tape	_mt0, _mt1, _mt2, ...
1/4 inch cartridge tape	_ct0, _ct1, _ct2, ...
Hydra Student Stations	_hd0, _hd1, _hd2, ...
Hydra Audio Stations	_ad0, _ad1, _ad2, ...
Parallel ports	_pp0, _pp1, _pp2, ...
Memory disk	_md0
Pipes	Unspecified

2. The drive ID of the device. The drive ID is an identifier that the device driver needs to access the device.

3. The name of the device driver for the device.

The following charts list the devices, drive IDs, and device drivers each WICAT system:

=== System 140 ===

Controller/Description	Recommended Devicename	Drive id	Device Driver	Possible Drive types
Scsi disk (the hard disk drive 0)	_dc0	0a0	scsi$140	winl2 winl9
Scsi disk (5 1/4 inch diskette)	_dx0	0b0	scsi$140	flop09a flop09b
Memory disk	_md0	0a0	mdsk$140	N/A
Serial port (the built-in terminal and the printer port)	_tt0.._tt1	0a0..1a0	tty$140	N/A
Centronics compatible parallel port	_pp0	0a0	cent$140	N/A
The print queue device	_pq0.._pq10	0a0..10a0	que$140	N/A
The null device (This should not be in the device table)	_null	0a0	null	N/A
Pipe devices (these should not be in the device table)	Unspecified	N/A	pip02$140	N/A

=== System 150/155/160 ===

Controller/Description	Recommended Devicename	Drive id	Device Driver	Possible Drive types
wd3 (also called wfc) 5 1/4 inch winchester drive 0..3	_dc0.._dc3	0a0..3a0	wd3$156	win12 win19 win30 win43
wd3 (also called wfc) 5 1/4 inch diskette drive 0..3	_dx0.._dx3	0b0..3b0	wd3$156	flop09a flop09b
wd2 (also called piggy-back) 5 1/4 inch winchester drive 0..3	_dc0.._dc3	0a0..3a0	wd2$156	win12 win19 win30 win43
wd2 (also called piggy-back) 5 1/4 inch diskette drive 0..3	_dx0.._dx3	0a0..3a0	wdmf$156	flop09a flop09b
I/O board 0. serial port 0..6	_tt0.._tt6	0a0..6a0	tty$156	N/A
I/O board 0. parallel port.	_pp0	0a0	cent$156	N/A
I/O board 1. serial port 0..6	_tt8.._tt14	0a1..6a1	tty$156	N/A
I/O board 1. parallel port.	_pp1	0a1	cent$156	N/A
ICI board 0, serial ports 0..7	_tt0.._tt7	0a0..7a0	tty$156	N/A
ICI board 1. serial ports 0..7	_tt8.._tt15	0a1..7a1	tty$156	N/A
Dei cartridge tape drive 0..3	_ct0	0b0..3b0	adei$156	N/A
Cipher magnetic tape drive 0..3	_mt0.._mt3	0a0..3a0	ciph$156	N/A
Smd disk controller board 0, drive 0..3	_ds0.._ds3	0a0..3a0	smd$156.dsr	smd84b smd168b smd474b
Smd disk controller	_ds4.._ds7	0a1..3a1	smd$156.dsr	smd84b

board 1, drive 0..3				smd168b smd474b
The memory disk	_md0	0a0	mdsk$156	N/A
The print queue device drive 0..10	_pq0.._pq10	0a0..10a0	que$156	N/A
The null device (not in device table)	_null	0a0	null	N/A
Pipe devices (not in device table)	Unspecified	N/A	pip02$156	N/A

=== System 100/200/220/300 ==

Controller/Description	Recommended Devicename	Drive id	Device Driver	Possible Drive types
Serial port 0..3 on the cpu board	_tt0.._tt3	0a0..3a0	tty$100	N/A
Serial port 0..3 on the port expander board	_tt4.._tt7	4a0..7a0	tty$100	N/A
Serial port 0..15 on the intelligent port expander board (ipe) 0	_tt8.._tt23	0a1..15a1	tty$100	N/A
Parallel port 0 on the ipe board 0	_pp0	0a1	tty$100	N/A
Serial port 0..15 on the ipe board 1	_tt24.._tt39	0a2..15a2	tty$100	N/A
Parallel port 0 on the ipe board 1	_pp1	0a2	tty$100	N/A
Dei cartridge tape drive 0..3	_ct0.._ct3	0b0..3b0	adei$100	N/A
Cipher magnetic tape drive 0..3 board 0	_mt0.._mt3	0a0..3a0	ciph$100	N/A
Cipher magnetic tape drive 0..3 board 1	_mt4.._mt7	0a1..3a1	ciph$100	N/A
Smd disk controller board 0, drive 0..3	_ds0.._ds3	0a0..3a0	smd$100.dsr	smd84b smd168b smd474b
Smd disk controller board 1, drive 0..3	_ds4.._ds7	0a1..3a1	smd$100.dsr	smd84b smd168b smd474b
IMI disk controller board 0, drive 0..3	_di0.._di3	0a0..3a0	imi$100.dsr	imi20 imi40
The intelligent floppy interface (flnt) 5 1/4 inch diskettes	_dx0.._dx3	0b0..3b0	flnt$100	flop09a flop09b
The intelligent floppy interface (flnt) 8 inch diskettes	_df0.._df3	0a0..3a0	flnt$100	flop015

The intelligent floppy interface (flnt) 5.25 inch winchesters	_dc0.._dc3	0d0..3d0	flnt$100	win12 win19 win30 win43
Hydra audio devices 0..29	_ad0.._ad29	0a0..29a0	audio$100	N/A
Hydra terminal devices 0..29	_hd0.._hd29	0a0..29a0	hydra$100	N/A
The memory disk	_md0	0a0	mdsk$100	N/A
The print queue device drive 0..10	_pq0.._pq10	0a0..10a0	que$100	N/A
The null device (not in device table)	_null	0a0	null	N/A
Pipe devices (not in device table)	Unspecified	N/A	pip02$100	N/A

If the device is a disk, there is an additional field in the DEVCONFIG.nnn.1 record that contains the drive type. The following chart lists all WICAT-supported drive types:

Drive Type	Description
FLOP09a	0.9 meg unformatted 4 sector 5 1/4" floppy (Standard 5 1/4" floppy)
FLOP09b	0.9 meg unformatted, 5 sector 5 1/4" floppy
FLOP015	1.5 meg unformatted 8" floppy
WIN12	12 meg unformatted winchester (Previously called CMI 10)
WIN19	19 meg unformatted winchester (Previously called CMI 15)
WIN30	30 meg unformatted winchester
WIN43	43 meg unformated winchester
SMD84b	84 meg unformatted smd with 1024 byte sectors
SMD168b	168 meg unformatted smd with 1024 byte sectors
SMD474b	474 meg unformatted smd with 1024 byte sectors
IMI20	20 meg formated imi disk
IMI40	40 meg formatted imi disk

If the device is a terminal. there are four optional fields that affect log ons on that terminal. (See the chapter on System Security and User Accounts. Part 2 in this manual or the SYSPROF Command description in the User Reference Manual for details.)

Note that only version 1 of DEVCONFIG.nnn is recognized by the WMCS. Confusion may result if you inadvertently create multiple copies of this file.

14.2.4 Establishing A Turnkey System

These are two recommended ways of creating a turnkey environment:

1. Modify the LOCALUP.COM file so that the turnkey environment is initiated whenever the system is booted. The following command line character strings might appear in LOCALUP.COM to establish an OIS environment whenever the system is booted:

   ```
   > cd sys$disk/ois/
   > &/ois <_tt1 >_tt1 ^ tt1
   > &/ois <_tt2 >_tt2 ^ tt2
   ```

   Thus. whenever the system is booted, a copy of OIS is initiated on terminals _TT1 and _TT2.

2. Use the USERPROF Command to modify UAF.DAT so that the turnkey application is initiated whenever the user logs on. For example, on the line in the UAR that asks for "command line," you could specify /OIS/OIS.EXE as the name of that user's interface with the system.

14.2.5 The System Command Files

As system manager, you are responsible for the system command files described in the chapter on How to Monitor and Customize System Use.

These command files provide a mechanism for configuring the environment of your system. Pay special attention to the DEVICEUP.COM, LOCALUP.COM, LOCALON.COM, and LOCALOFF.COM command files.

14.2.6 Dynamic Configuration Of Devices

The MNT Command makes it easy to include a device as part of your system's configuration while the system is running.

Floppy-disk drives. tape drives. terminals, printers. etc., can all be mounted and dismounted without having to rebuild the WMCS.

The DSTAT Command can also be used to modify device characteristics.

14.2.7 Non-WICAT Disk Devices

To facilitate customers who want to add non-standard disk devices to WICAT disk controllers (especially 5.25-inch Winchester and SMD devices), there is a disk configuration file that can be used to supply device characteristics without having to modify the device driver.

This process should only be attempted by knowledgeable system programmers.

The source for the disk configuration is in /SYSINCL.SYS/DISKCFG.ASM. To add a new type of disk to your system, follow these steps:

1. Edit /SYSINCL.SYS/DISKCFG.ASM. Create a new section for the new disk type. Be very careful to follow the format used for other drives defined in the file.

2. Exit the editor. You now have a source module with all previously defined drive types. plus your new type.

3. Assemble the DISKCFG.ASM. Use the ASM assembler. Note that there are includes in the source, so the assemble step must use the INCL Command. For example:

```
> incl diskcfg.asm diskcfg.tmp
> asm diskcfg.tmp :L
> del diskcfg.tmp :auto :nolog
```

4. If the file assembled without errors, then link the object file. e.g., > **link diskcfg.**

5. Run MAKEDSR :SIMPLE on the resulting image file, e.g., > **makedsr diskcfg.exe disk.cfg :simple.**

6. Put the resulting file in the /SYSDSR/ directory, e.g., > **ren disk.cfg sysdsr/.**

All of the standard utility programs use this file as needed. For instance. DINIT reads it to get drive parameters as specified by the drive type. SYSPROF reads it to get the names of all drive types. BTUP reads it to obtain drive parameters.

14.2.8 Specify The Number Of TTY Ports

The number of TTY ports you specify when executing the SYSPROF Command does not pertain to IPE and ICI boards.

Specify 8 TTY ports for a System 100, 200, 220, or 300.

Specify 3 for a System 150-1WS, or 150-3WS.

Specify 7 for a System 150-6WS, 155, or 160. Specify 14 if your system has two I/O boards.

CHAPTER 15

CRASH RECOVERY

15.1 AUDIENCE

The information in this chapter is written for both beginning and experienced system managers.

15.2 INTRODUCTION

The term "crash" describes the abnormal termination of either a process or of the system itself. For example, when the power goes out while the system is running, the system crashes. If a program executes an illegal instruction, that process crashes.

Even though the WICAT system has been carefully tested (both its hardware and software), and many safeguards have been built into your system to avoid catastrophic crashes. you should be aware that the system can go down from time to time. You should therefore prepare for system crashes by having a well-defined backup procedure.

15.3 KINDS OF CRASHES

Generally speaking, there are two kinds of crashes:

1. Those where no data are lost.

   Note that the loss of data is not always apparent.

2. Those resulting in loss of data.

Execute the RECOVER Program (described in the WICAT Multi-user Control System (WMCS) User Reference Manual) to recover from a system crash (described later in this chapter). RECOVER reconstructs the file system on a disk and thereby endeavors to retrieve as much data as possible that may have been lost.

The following sections describe some common kinds of crashes, what to do in the event of a particular kind of crash, and what damage (if any) to anticipate as a result of the crash.

15.3.1 Process Crash

A process crash occurs when a single program terminates abnormally. This kind of crash is usually inconsequential to the system itself. It may, however, have serious consequences for the process. For example, if the text editing program crashes, all changes in the buffer copy of the file since it was last saved will be lost.

After a process crashes, the terminal usually displays these characteristics:

1. The terminal still responds to commands.
2. Either a stack dump appears on the screen or the right angle bracket (the CIP prompt) appears on the screen unexpectedly.

Process crashes can occur for any number of reasons, but they often occur because of a bug in the program being used, e.g., the program accessed memory that was not assigned to it, executed a divide-by-zero operation, performed an illegal instruction, etc. Frequently, the bug is due to invalid user-supplied data, e.g., the operator types something that the program does not expect.

Hardware failure, such as a memory parity error, is another cause of process crashes.

The WMCS has little or no control over these kinds of crashes, but it does attempt to detect and report such errors. When an error is detected by the WMCS, a diagnostic message, or stack dump, is written to the file designated as SYS$ERROR for the process. This diagnostic message specifies the kind of error involved; the processor status, i.e., program counter, stack pointers, etc.; a stack dump of the last few words on the top of the stack; and a register dump. The data displayed are designed to help the programmer, who wrote the crashed program, discover the location and the nature of the error.

Depending upon the circumstances, you may want to report the process crash to the responsible programmer or organization. Use the "System Log/Diagnostics" form in Appendix A to record the information you need to report.

The most important pieces of information to note from the stack dump are:

1. The diagnostic message (usually the first line of the report).

2. The process name.

3. The program counter (PC).

There is no standard recovery procedure following a process crash.

15.3.2 Single-terminal Hang

A single-terminal hang is a crash that occurs when one terminal on the system becomes inoperative, i.e., the cursor freezes and does not respond to keyboard input. In general, this kind of crash is not detrimental to the system.

To determine whether the crash is a single-terminal hang (rather than a system hang), check other terminals connected to the system. If the other terminals are functioning, you have a single-terminal hang.

These are some possible causes for single-terminal hangs:

1. A program has gone awry and will not terminate.

2. A cable is disconnected.

3. The terminal has temporarily become inoperative.

4. The terminal has a hardware failure.

Use the following suggestions to attempt to recover from a single-terminal hang:

1. Check the cable to make sure its connection to the terminal and the serial port is solid.

2. If the hung terminal is n<u>ot</u> TT0 on a System

140 or System 150 (the terminal for the system manager) turn the terminal off and then on again. Then strike [RETURN] to see whether the terminal responds.

If the hung terminal is _TT0 on a System 140 or System 150, do not turn the terminal off and then on (doing so reboots the system). Follow this procedure instead:

a. Press [SET-UP].

b. Press the spacebar once to take the terminal off line.

c. Press [SET-UP] again.

d. Type [ESC] c.

e. Press [RETRN] to see whether the terminal responds.

3. If the terminal does not respond, execute the KILL Command from another mounted terminal. In most cases. this is sufficient to make the terminal usable again.

4. If killing the process does not work, execute the DMNT Command from another mounted terminal to dismount the hung terminal. Then execute the MNT Command to remount the terminal.

5. If the foregoing suggestions are ineffectual, ask the other users to log off. Then execute the SHUTDOWN Command, specifying the :REBOOT Switch.

If none of the foregoing solves the problem, the terminal may have a hardware problem. Report the error to WICAT Field Service (if it is a WICAT terminal).

15.3.3 System Hang

A system hang results when the entire system crashes, i.e., none of the terminals on the system respond. This error could—but does not usually—cause damage to the system's file structure. A system hang usually occurs with no sign of an error: the cursor simply freezes on all the terminal screens and the system stops responding. Sometimes a stack dump appears on the screen.

To determine whether the crash is a system hang, check all mounted terminals. If _TT0 and the other terminals do not respond to keyboard input, the entire system has crashed.

Follow these steps to recover from a system crash:

1. Use the Reset Control to reboot the system.

2. Execute the RECOVER Program to check and repair any damage to the file system (read the description of RECOVER in the WICAT Multi-user Control System (WMCS) User Reference Manual).

If the symptoms of this kind of crash are repeated regularly, contact WICAT Field Service.

15.3.4 Accidental System Reset

Pressing the reset control without properly shutting down the system terminates all processes on the system.

Follow this procedure if the system is accidentally rebooted:

1. Allow the boot, initiated by the Reset Control, to finish.

2. Execute the RECOVER Program to check and repair any damage to the file system that may have occurred.

Setting the autoflush flag (read the BTUP Command description) on all disks, or running the LOGFLUSH program minimizes the risks associated with this kind of crash.

15.3.5 Power Failure

The effect of a power failure is similar to that of an accidental system reset in that all processes are terminated.

Follow these steps in the event of a power failure:

1. Turn the Power Control to the OFF position

before the power comes back on.

2. When power returns. wait until you feel certain that power is stable before you turn the system on.

3. Execute the RECOVER Program.

Setting the autoflush flag on all disks, or running the LOGFLUSH Program minimizes the risks associated with this kind of crash.

15.3.6 Equipment Failure

The system can fail as the result of a hardware malfunction.

The following symptoms may or may not appear in the case of an equipment failure:

1. A stack dump may appear on the screen of the terminal running the crashed process.

2. A parity error may appear on the screen of the terminal connected to serial port TT0.

3. The process may hang when it attempts to access a device.

4. CRC and other device errors may appear on the screen.

The effects of equipment failure are varied, but in most cases little or no damage will have occurred to the file system. When the equipment has been repaired, use the RECOVER Program to check and repair any damage to the file system.

CHAPTER 16

THE SYSTEM DISK

16.1 TAPE AS A SYSTEM DEVICE

Using a tape as the system device can be cumbersome. If the files on the tape are not placed in a convenient order, much time is wasted searching for them.

When you boot from a tape. your primary goal should be to get enough files onto the disk so that you can execute from the disk instead of the tape.

Files that will be executed regularly should be close to the front of the tape so that you can rewind the tape and get to them quickly. The files following the CIP executable file should be placed in this order:

1. /SYSEXE/DINIT.EXE — If the system disk is bad, you may want to reinitialize it.

2. /SYSDSR/<diskdriver>.DSR — Required by DINIT.

3. /SYSEXE/MNT.EXE — You will want to mount the disk as soon as possible.

4. /SYSDSR/<diskdriver>.DSR — This is required by MNT.

5. /SYSEXE/CRD.EXE — Used to create the SYSEXE directory on the disk device.

6. /SYSEXE/COPY.EXE — Needed to copy files from the tape to the disk.

7. /SYSEXE/COPY.EXE — One of the most important programs to have on the disk is the COPY Command. It is placed after the other COPY so it can be copied without rewinding the tape.

8. /SYSEXE/DIR.EXE — It is important to have the DIR Command near the front of the tape.

9. The /SYSEXE/ directory

10. All files from /ROOTDIR/

11. Other directories

The following is an example of how you would use a tape to boot the system, then mount the hard disk, and assign it as the system disk:

1. You boot from the tape device because something has gone wrong with the system disk.

2. The tape is positioned just before the DINIT Command.

3. You would rather not reinitialize the disk unless you have to. so you use the MNT Command to try to mount the system disk. The tape is repositioned past DINIT and the first disk driver, loads the MNT Command, and uses the driver following the MNT Command. The tape is then positioned just before the CRD Command.

4. If the disk is successfully mounted, you assign it the logical name of SYS$DISK. From then on you are executing from the disk, and you can find the source of the problem (a missing or bad file) and fix it.

5. If you cannot mount the disk, you will have to reinitialize it. To do so. use the REW Command to rewind the tape so that the DINIT Command is ahead of your position on the tape. Then use the DINIT Command to initialize the disk. The driver following the DINIT Command will also be used, and you will be positioned just before the MNT Command.

6. Next, mount the disk using the MNT Command and the driver following the MNT Command. You will then be positioned just before the CRD Command.

7. Execute the CD Command (requires no files from the tape) to set the default device and directory to /ROOTDIR/ on the disk.

8. Enter the CRD Command to create the /SYSEXE/ directory on the disk. The tape will be positioned just before the COPY Command.

9. Execute a CD Command (requires no files from the tape) to set the default device and directory to /SYSEXE/ on

the disk.

10. Execute the COPY Command to copy all files from the /SYSEXE/ directory on the tape to the /SYSEXE/ directory on the disk.

11. Assign the logical name SYS$DISK to the disk, whereupon you can execute from the disk without having to worry about the order in which the commands are located on the tape. You can also copy the WMCS files and other standard files from the tape to the disk.

CHAPTER 17

THE MEMORY DISK

The MDSK$nnn.DSR device driver allows the WMCS to use a portion of main memory as though it were a disk; that is, the device driver interacts with memory and responds to the WMCS as though a portion of memory were made up of sectors (this is not the same as disk cache).

Mount the memory disk as you would any other device. You can use it as you would any other disk, e.g., assign it as your default device, create files and directories on it. check its sectors, etc.

When new sectors are allocated to a file, the driver allocates more of the system's memory to the disk. When files are deleted or shortened, the driver deallocates the memory to which they were assigned.

Note that if the system is shut down or crashes, any files or directories on the memory disk are lost.

The memory disk occupies a minimum of 4 kilobytes of memory and has a 2 megabyte capacity.

Chapter 18

Networking

The material in this chapter is for inexperienced and experienced system managers.

Networking allows processes to access resources on other computers. These resources can be files, devices, processes, memory, etc. In fact, almost anything which can be done in WMCS locally can be done remotely with another computer. A process does not need to do anything special in order to use networking; it is performed automatically by the WMCS.

There are certain duties which the system manager must perform in connection with networking. They are basically just an extension of duties already being performed. They include:

1. Authorizing remote users of the system.
2. Specifying system network configuration information.
3. Enabling networking on the system.
4. Maintaining system security and integrity.

System Identification

Networking requires a method for identifying individual systems on the network. This is accomplished with two different, but related, system identifiers assigned by the system manager. They are the nodename and the site ID number. Both system identifiers <u>must</u> be unique with respect to the nodenames and site IDs of the other systems on a network.

The nodename will be used most frequently by users, such as when referencing files, devices, systems, etc.

An example of copying a file from the default directory to a 9-track tape on another system named "__WALLY" is:

```
> copy sample.txt __wally_mt0/*/
```

The site ID will be used most frequently by application programs. There are WMCS supervisor calls (SVCs) which allow conversion between a nodename and site ID and allow an application program to get the site IDs of all the systems on a network. See the WMCS Programmer's Reference Manual for more details on network-oriented supervisor calls.

Nodename

The first system identifier is the "nodename" of a system. It may contain up to 16 alphanumeric characters, tildes or dollar signs which are optionally preceded by two underscore characters. An example is "__MEPH". If a nodename has not been defined for a computer, the default supplied at boot time is __NODE.

Nodenames can be used with devicenames to uniquely identify a device on any system in a network. For instance, assume your nodename is __BOB. The name __BOB_TT0 uniquely identifies the device _TT0 on system __BOB anywhere in the network. If a devicename is specified with no nodename prepended to it, the default nodename is the nodename of the computer on which a process is executing. Therefore, a program that is executing on BOB can open the device _TT0 on BOB by specifying _TT0 or __BOB_TT0. However, a program executing on the computer GARY would have to specify __BOB_TT0 in order to open the device _TT0 on BOB. In all WMCS SVCs and utilities, a nodename is accepted as part of a devicename.

For example, a program on BOB can open a file on GARY named SYS$DISK/EXAMPLE/TEST.DAT by executing the _open SVC with the filename __GARY_SYS$DISK/EXAMPLE/TEST.DAT. The _open SVC returns a LUN which the program can then use for all subsequent I/O to the file. The entire network operation is transparent to the program. The program does not need to know it really opened a file on a remote computer. The only difference between opening a local file versus a remote file is the filename specification.

> NOTE: The nodename of a computer must be unique within any given network. If two nodes in a network have the same nodename, one of them will not be accessible.

The nodenames are exchanged dynamically when a physical network device is mounted. This nodename exchange takes place approximately five to ten seconds after a new computer is added to a network (when a new physical network device is mounted).

Site ID Number

The second system identifier is the system identification number, or site ID. It is a 16-bit number which is used by the network code to uniquely identify a computer. The system manager may use the numbers from 1 to 1024 (hex 0001-0400) as site IDs. The number 0 is reserved and cannot be used. WICAT has also reserved the numbers 1025 to 65536 (hex 0401-FFFF) to be assigned by WICAT Customer Service. If a system manager wants a site ID for his computer that no other system will ever use, he can call Customer Service and get a unique number allocated to him.

The site ID appears as the first four hexadecimal digits of a process ID number (the high-order 16 bits). In a PSTAT display, the left four digits of the PID are the site ID of the computer on which the PSTAT is executing. On all SVCs which use a PID, if the high-order 16 bits of the PID are zero, the site ID of the current computer is used. For example, if a computer has the site ID 0006 and a process number 0243 exists on that computer, the user can refer to that process with the PID 0006024c or the PID 0000024c. However, if the PID 00000243 is used on a different computer with site ID 0008, it really refers to PID 00080243.

Like the site ID portion of the PID, the site ID can be specified as 0000 on all SVCs which have a site ID parameter. If a zero is passed in for a site ID parameter, the actual site ID used is the site ID of the computer on which the SVC is executed.

The _sidlst SVC returns a list of the site IDs for all systems which are up and running on the network. The first site ID in the returned list will always be the site ID of the system on which the SVC was executed.

Network Devices

Networking introduces a different kind of device: the network device. This class of devices includes physical network devices, such as the Ethernet and Arcnet boards, and "virtual circuit" devices. When a process must communicate with another system, it does so through a "virtual circuit," which the networking system software creates and manages. There may be more than one virtual circuit per physical network device.

A virtual circuit is a "network" class device that is used much like other WMCS devices. Its function is similar to a TTY device with a modem on it. The user mounts a virtual circuit, does a dial or answer on it, and then reads and writes data to it. Virtual circuits always have a device mnemonic of "_VCxxx", where "xxx" is the device number. There may be at most 256 virtual circuits mounted and open on a single system.

NSP

A Network Surrogate Process (NSP) is responsible for mounting and tending to the virtual circuits. This program must be executing in order for networking to operate. NSP automatically mounts virtual circuit devices as it needs them, based on a user-specified range of device numbers (maintained by NSYSPROF). Then NSP performs the requested operations locally in behalf of another process on another system. The NSP process must be forked when the system is booted. After that, the process clones itself so there is one always available for new remote processes to use.

The NSP image file SYS$DISK/SYSEXE/NSP.EXE must be installed at boot time with the CHNGSUPER privilege. This is necessary because it must do some things in processor supervisor mode.

The operating system and the NSP work extremely close together in order to make networking happen. When a local process attempts to do something on a remote sytem over the network, the operating system "calls" the remote system through an available virtual circuit. The NSP process on the remote system "answers" the call. The NSP then performs supervisor calls on behalf of the local process for as long as the process exists or the virtual circuit is open. As soon as the local process dies or a "hangup" is performed on the virtual circuit by the process, the NSP on the remote system dies.

The first NSP to execute sets its process name to "NSP_answer," its execution priority to zero and its timeslice to 100 milliseconds. Its port (as displayed by PSTAT) is set to the name of the virtual circuit the NSP is using. When a remote process "calls" the system, the NSP clones itself and then changes its process name to "NSP_xxxxxxxx", where "xxxxxxxx" is the process ID of the remote process it is serving. It also changes its execution priority, timeslice and privileges to those specified in the network user authorization file. The newly cloned NSP sets its process name to "NSP_answer" and waits for a remote process to "call" it.

Authorizing Remote Users

Networking will allow processes on other systems to access system resources. Like the USERPROF program which is used to authorize local users on the computer, the NUSERPROF program is used to authorize remote users on the local computer. This program assigns privileges, execution timeslice and priority, and applicable restrictions to the remote users. See the WMCS User's Reference Manual for instructions on how to operate NUSERPROF.

System Network Configuration

Just as a system must be configured with local devices for proper utilization of the devices, so too must a system be configured for proper utilization of networking. The /ROOTDIR/NETCONFIG.xxx file contains the configuration information for networking and virtual circuit devices and is created and maintained by the NSYSPROF utility. See the WMCS User's Reference Manual for instructions on how to operate the NSYSPROF utility.

The networking information includes the local system nodename and site ID. It also has information about buffer cache sizes for data packet reception and transmission and network overhead.

The nodename and site ID are specified by the system manager and maintained through the NSYSPROF utility.

The data packet buffer cache sizes can be modified in order to "fine tune" a system. The size of the data packet buffer caches depends upon the amount of available system memory and the amount of networking a system is expected to do. The more memory a system has, the more data packet buffers a cache may have. The more network traffic a system is expected to have, the more buffers it should have in order to increase the throughput of the system.

The network overhead buffer cache size depends upon the amount of network traffic a system will have and whether the system is a "gateway". A gateway is a system which has more than one physical network device, e.g., an Ethernet board and an Arcnet board. A gateway must handle all network traffic between systems connected to the physical network devices and therefore will have a greater networking load on the system. Increasing the amount of network overhead buffer cache helps to alleviate the increased load.

The NETCONFIG.xxx file also contains a restriction on the virtual circuits which NSPs are allowed to use. There are 256 virtual circuits named "_VC0 through "_VC255". The NSYSPROF utility allows the system manager to restrict NSPs to a specific subset of virtual circuits. An example would be "_VC10" through "_VC20". This would restrict the number of concurrent NSPs to a maximum of 11 virtual circuits, and therefore the number of remote users using the local system.

System Network Security

The NSYSPROF utility also maintains the security of the local system with respect to the network. It is important that a system on a network can be checked to make sure that it is really the system it says it is. System verification is performed using a "public/private key" encryption algorithm. The public/private keys are generated by the KEYGEN utility, or by a function of the NSYSPROF utility, NK, which starts up KEYGEN. NSYSPROF also maintains the public keys of all systems on the network in a file called /ROOTDIR/NETPUBKEY.xxx. The private key of the local system is contained in the NETCONFIG.xxx file.

Site ID Verification

The network must be able to prevent imposters from "pretending" they are a computer on your network. To illustrate the problem, consider the following situation:

1. There are three legal computers in a network: "A", "B", and "C".
2. Computer "C" contains sensitive information and computer "B" is granted certain privileges which allow him to access the files on computer "C".
3. A malicious computer "D" wishes to get access to the sensitive data on computer "C".
4. Computer "D" sets his own site ID and nodename to match those of computer "B".
5. Somehow malicious computer "D" attempts to get onto the network as computer "B" by breaking computer "B"'s access to the network by either turning "B" off, disconnecting the physical link which hooks "B" into the network, or, by getting on the network while computer "B" is not on the network.
6. Computer "D" then gets onto the network as if it were the real computer "B" and does whatever he wants with the files on computer "C" because "C" gives privileges to "B" and does not know that "B" is not really who he says he is.

There are other situations similar to the one above where one computer can attempt to imitate another computer. To solve this problem, optional site ID verification has been implemented. This verification procedure achieves the purpose of security at the expense of overhead to the initial communication between two computers. Because of this additional overhead, site ID verification is optional. It takes approximately ten seconds for two computers to verify that they are both who they say they are. This slow verification is done once when the first connection between the two computers is made. Thereafter, a fast re-verification is done on every subsequent connection (using information exchanged during the first verification) which makes sure that no imposters have broken into an already verified link. The fast re-verification does NOT add any noticeable overhead.

The site ID verification procedure is accomplished using a public/private key encryption algorithm. A pair of keys, a public key and a private key, is generated by a computer. The public key is distributed to every computer on the network. The private key is kept by the computer that generated it. It takes about one second to encrypt or decrypt 16 bytes of data using the public/private algorithm. During the verification procedure, 16 bytes are encrypted/decrypted eight times as information is exchanged back and forth between the two computers. This causes the ten seconds of overhead.

With public/private keys, data that is encrypted with a public key can only be decrypted with the corresponding private key, and data that is encrypted with the private key can only be decrypted with the corresponding public key. Thus, if a message can be decrypted with a public key, then the message had to have been generated by a computer that had the private key. Also, if a message is encrypted with a public key, only a computer with the private key can successfully decrypt it. This is often called an "electronic signature". The basis of this procedure is that the private keys are kept secret. In the example given at the start of this section, computer "D" would have to know what the private key of computer "B" is in order to successfully pretend it is the real "B".

The actual verification procedure between two computers, "A" and "B", is as follows:

1. Computer "A" generates a 16-byte random message "x".

2. Computer "A" encrypts the message "x" with the public key of computer "B". (Remember, only computer "B" can successfully decrypt the message.)

3. Computer "A" sends the encrypted message "x" to computer "B".

4. Computer "B" decrypts the message "x" with his private key and re-encrypts message "x" with the public key of computer "A".

5. Computer "B" generates a 16 byte random message "y".

6. Computer "B" encrypts the message "y" with the public key of computer "A".

7. Computer "B" sends messages "x" and "y" to computer "A", which is the only computer that can successfully decrypt them.

8. Computer "A" decrypts the message "x" with his private key and compares it to the original message "x" that he generated. If they are the same, then computer "A" knows that computer "B" has the private key of "B"; therefore computer "B" must be the real "B".

9. Computer "A" then decrypts the message "y" with his private key and re-encrypts the message with the public key of computer "B".

10. Computer "A" sends the message "y" back to computer "B", which is the only computer that can successfully decrypt the message.

11. Computer "B" decrypts the message "y" with his private key and compares it to the original message "y" that he generated. If they are the same, then computer "B" knows that computer "A" has the private key of "A"; therefore computer "A" must be the real "A".

The public/private key algorithm exchanges keys (messages "x" and "y") during the verification process which are used later for other, faster methods of encryption. All subsequent site ID re-verifications and DES or FAST encryption of user data use these keys exchanged during the initial verification.

If no site ID verification is done (no public/private keys are defined) the networking software still exchanges randomly generated keys that are encrypted with a default fast encryption algorithm rather than with the slow public/private algorithm. These keys are not as secure as the public/private algorithm.

> NOTE: Even if you do not use site ID verification on your computer, if any other computer in your network uses it, you must define public keys for that computer. This is done by adding a record for that computer (or node) to your NODCONFIG.xxx file with the NSYSPROF utility.

Keys are generated by the KEYGEN utility and are merged by the NSYSPROF utility. See the WMCS User's Reference Manual for descriptions of these utilities.

Network Encryption

There are two kinds of encryption that are supported on data that is written across the network. These two kinds are:

1. DES — National Bureau of Standards Data Encryption Standard.

2. FAST — WICAT's implementation of a polyalphabetic autokey algorithm.

WICAT's implementation of the DES algorithm is done totally in software, and can encrypt/decrypt slightly more than one Kbyte per second on an idle System 2220. It can encrypt/decrypt slightly less than one-half Kbyte per second on an idle System 150 or System 160. WICAT's implementation of the FAST encryption algorithm is approximately five times faster than the DES algorithm.

There are two ways that one or both of the above encryptions can be enabled on user data that is passed across the network. The first way to force encryption of user data is for the system manager to set one or more of the encryption flags in the NSYSPROF utility. When these bits are set in a remote system's record, then all calls placed from the local computer to the remote will use the specified form of encryption for both directions of data flow between the computers. However, Calls placed by that remote to the local may or may not encrypt, depending on the state of the flags in the remote computer's NSYSPROF files.

The second way to cause encryption to take place on user data is enabled through the use of two process attribute bits. These two bits are part of a 16-bit "bitmask" that is now passed in on a create process SVC as the top 16 bits of the create process mode parameter. If a process that has these bits set causes the network to access a remote computer, then all data that is written or read over the network by that process will be encrypted using the specified method. With the USERPROF utility, the system manager can set up a default state for these encryption bits which will be assigned to the user's CIP (or other default process) when a user logs on.

Examples of How to Use the Network

These examples assume that nodenames and site IDs have been assigned, necessary machine verification keys have been generated, and users have been authorized. The sample network consists of four computers that have been assigned nodenames of __A,__B, __C, and __D.

> NOTE: The leading two underscores before the nodenames are optional in these examples.

Accessing Remote Files

1. Assume that the sys$disk on "A" is _ds0, and the default directory is _ds0/syshlp on computer "A". The following commands all do the same thing. They copy the file vew.hlp from directory _ds0/syshlp on computer "A" and put it on _dc0/syshlp on computer "B".

```
> copy vew.hlp  __b_dc0/syshlp/vew.hlp
> copy syshlp/vew.hlp  __b_dc0/*/vew.hlp
> copy sys$disk/syshlp/vew.hlp  __b_dc0/syshlp/*
> copy ds0/syshlp/vew.hlp  __b_dc0/syshlp/*
> copy __a_sys$disk/syshlp/vew.hlp  __b_dc0/*/*
> copy __a_ds0/syshlp/vew.hlp  __b_dc0/syshlp/*
```

2. Assume the same environment as example 1. The following commands copy the file from computer "B" back to "A".

```
        > copy __b_dc0/syshlp/vew.hlp *
        > copy __b_dc0/syshlp/vew.hlp syshlp/vew.hlp
        > copy __b_dc0/syshlp/vew.hlp sys$disk/*/*
        > copy __b_dc0/syshlp/vew.hlp ds0/*/vew.hlp
        > copy __b_dc0/syshlp/vew.hlp __a_sys$disk/syshlp/*
        > copy __b_dc0/syshlp/vew.hlp __a_ds0/syshlp/vew.hlp
```

3. Assume that the default directory is on computer "A". The following commands TYPE the file test.dat from _ds0/tmp on computer "A".

```
        > type tmp/test.dat
        > type ds0/tmp/test.dat
        > type __a_ds0/tmp/test.dat
```

4. Assume that sys$disk on computer "A" is _ds0. The following commands copy all files from "sys$disk/tmp" on all computers in the network to the directory _ds0/example on computer "A". (This is an example of node wildcarding.)

```
        > copy __*_sys$disk/tmp/* __a_ds0/example/*
        > copy __*_sys$disk/tmp/* __a_sys$disk/example/*
```

5. The following command will VEW the file garb.dat on computer "D" in its "sys$disk/example" directory.

```
        > vew __d_sys$disk/example/garb.dat
```

6. This example shows how logical names can be used on the local computer. The following commands assign a logical name and TYPE the file vew.hlp from sys$disk/syshlp on computer "D".

```
        > ddd := __d_sys$disk
        > type ddd/syshlp/vew.hlp
```

Accessing Remote Devices

To access a remote device, the user must TYPE the name of the device prepended with the nodename of the computer which owns the device. Node wildcarding allows the user to access multiple devices on multiple computers with the same command.

> NOTE: If a nodename is left off of a devicename, the nodename will default to be the node on which the program is executing.

For example, if a program is running on computer "B", then both __b_dc0 and _dc0 refer to device dc0 on computer "B".

1. The following command sends the message "hello" to all terminals on all computers on the network. (This is an example of node wildcarding.)

   ```
   > send __*_* "hello"
   ```

2. The following commands do a DSTAT display of device TT3 on computer "D" and TT2 on computer "B". Assume DSTAT is being executed on computer "B".

   ```
   > dstat __d_tt3,tt2
   ```

3. The following command does a DSTAT of TT3 on computers "A" and "B".

   ```
   > dstat __[ab]_tt3
   ```

4. The following command do a DEV of all disk devices on all computers in the network.

   ```
   > dev *_* :class=disk
   ```

5. The following command formats and initializes disk dx0 on computer "C" and gives the disk the label of "EXAMPLE". (If executing DINIT on computer "C", the nodename "__c" is optional.)

   ```
   > dinit __c_dx0 :format "EXAMPLE"
   ```

6. The following command mounts device _mt0 on computer "B" with 3200 BPI. (If executing MNT on computer "B", the nodename "__b" is optional.)

   ```
   > mnt __b_mt0 :density=3200
   ```

7. The following command will TYPE the file sys$disk/syshlp/vew.hlp on computer "C" with output redirected to _TT0 on computer "D".

   ```
   > type __c_sys$disk/syshlp/vew.hlp >__d_tt0
   ```

Running Processes on Remote Computers

To run processes on a remote computer the system manager must have given the UIC crproc privileges using the NUSERPROF utility. There are several utilities which only work on the computer they are executed on, for example, PSTAT, DM, and MEMTEST. To get a PSTAT or DM, etc., of a different computer, these programs must be executed on that computer.

The following are examples of how to use the CIP create process syntax to do this. In these examples, assume that the user has a CIP executing on computer "A".

1. The following commands will do a stat of "A".

```
> stat
> {a}stat
```

2. The following command will do a stat :full of "B".

```
> {b}stat :full
```

3. The following commands will fork a DM on computer "C" with input, output, and error all being redirected to device _TT0 on computer "D".

```
> {c}&dm <>^__d_tt0
> &{c}dm <>^__d_tt0
```

4. The following does the same as example 3 above, except all data that is written over the network will be encrypted using the DES algorithm.

```
> {c :desencrypt}&dm <>^__d_tt0
```

5. The following command will VEW the file dc0/tmp/garb.dat on computer "D" and VEW will be executing on computer "D", with all terminal I/O going to the user's logon terminal on computer "A". (Without the {d}, VEW would execute on computer "A" using the same remote file.

```
> {d}vew __d_dc0/tmp/garb.dat
```

6. The following is an example of some of the complex things that can be done with remote executes and pipes. Assume that the user is logged on with a CIP running on computer "A". In this example the image file type.exe is loaded from /sysexe/ on computer "B" and executed on computer "D". It will TYPE the file ds0/tmp/garb.dat from computer "A", with the output of the TYPE going to a pipe which will be automatically mounted on computer "D". A program DISPATCH will be loaded from /sysexe/ on computer "C" and executed on computer "C" which will read the output from the pipe on computer "D" and which will then output the data to the user's logon terminal on computer "A" and also to "ttl2" on computer "B".

```
> {d}__b_dc0/sysexe/type __a_ds0/tmp/garb.dat | {c}dispatch __b_ttl2
```

Default Directories

With networking, the user can change his default directory to be any directory on any directory device in the network. The following are some examples of what can be done.

In these examples, assume that the user logged on to computer "A", and his default directory starts out as __a_ds0/example/.

1. The following commands both VEW the file __b_dc0/example/file.dat.

```
     a.  > vew __b_dc0/example/file.dat

     b.  > cd __b_dc0/example
         > vew file.dat
```

2. The following commands both TYPE the file __b_dc0/example.test.dat.

```
     a.  > type __b_dc0/example/test.dat

     b.  > cd __b_dc0/example
         > type test.dat
```

3. The following commands both TYPE the file __b_dc0/syshlp/vew.hlp. This example illustrates that if a directory name is given without a devicename, the device defaults to be the device containing the default directory — no matter what node that device is on.

```
     a.  > type __b_dc0/syshlp/vew.hlp

     b.  > cd __b_dc0/example
         > type syshlp/vew.hlp
```

APPENDIX A

FORMS

Use the forms in this appendix to simplify system administration. This appendix contains the following forms designed to help you manage your system: 1. List of User Groups and Users 2. Ports and Peripheral Codes 3. System Log/Diagnostics Reproduce this form and use it to record power failures, hardware failures, system crashes, etc.

A.1 LIST OF USER GROUPS AND USERS

Group	Function
1	System manager
2	
3	
4	

Group 1 User no.	Username	UIC User no., Group no.
1	SYSTEM	[0001,0001]
2		[0002,0001]
3		[0003,0001]
4		[0004,0001]
5		[0005,0001]
6		[0006,0001]
7		[0007,0001]
8		[0008,0001]
9		[0009,0001]
10		[0010,0001]

Group 2 User no.	Username	UIC User no., Group no.
1		[0001,0002]
2		[0002,0002]
3		[0003,0002]
4		[0004,0002]
5		[0005,0002]
6		[0006,0002]

7		[0007,0002]
8		[0008,0002]
9		[0009,0002]
10		[0010,0002]

Group 3 User no.	Username	UIC User no., Group no.
1		[0001,0003]
2		[0002,0003]
3		[0003,0003]
4		[0004,0003]
5		[0005,0003]
6		[0006,0003]
7		[0007,0003]
8		[0008,0003]
9		[0009,0003]
10		[0010,0003]

Group 4 User no.	Username	UIC User no., Group no.
1		[0001,0004]
2		[0002,0004]
3		[0003,0004]
4		[0004,0004]
5		[0005,0004]
6		[0006,0004]
7		[0007,0004]
8		[0008,0004]

FORMS

9 ______________________________ [0009,0004]

10 ______________________________ [0010,0004]

A.2 PORTS AND PERIPHERAL CODES

SYSTEM 140

Port Label	Peripheral Code	WMCS Devicename
__________	______________	_TT0
P1	______________	_TT1

SYSTEM 150/155/160

I/O or ICI Board No.1

Port Label	Peripheral Code	WMCS Devicename
P0	________________	_TT0
P1	________________	_TT1
P2	________________	_TT2
P3	________________	_TT3
P4	________________	_TT4
P5	________________	_TT5
P6	________________	_TT6*
P7	________________	_TT7*

I/O or ICI Board No. 2

Port Label	Peripheral Code	WMCS Devicename
P0	________________	_TT8
P1	________________	_TT9
P2	________________	_TT10
P3	________________	_TT11
P4	________________	_TT12
P5	________________	_TT13
P6	________________	_TT14*
P7	________________	_TT15*

*Available only on ICI Boards

SYSTEM 100/200/220

CPU Ports

Port Label	Peripheral Code	WMCS Devicename
P0	________________	_TT0
P1	________________	_TT1
P2	________________	_TT2
P3	________________	_TT3

Port Expander

Port Label	Peripheral Code	WMCS Devicename
P0	________________	_TT4
P1	________________	_TT5
P2	________________	_TT6
P3	________________	_TT7

IPE Board No. 1

Port Label	Peripheral Code	WMCS Devicename
P0	________________	_TT8
P1	________________	_TT9
P2	________________	_TT10
P3	________________	_TT11
P4	________________	_TT12
P5	________________	_TT13
P6	________________	_TT14
P7	________________	_TT15
P8	________________	_TT16
P9	________________	_TT17

FORMS

P10		_TT18
P11		_TT19
P12		_TT20
P13		_TT21
P14		_TT22
P15		_TT23

IPE Board No. 2

Port Label	Peripheral Code	WMCS Devicename
P0		_TT24
P1		_TT25
P2		_TT26
P3		_TT27
P4		_TT28
P5		_TT29
P6		_TT30
P7		_TT31
P8		_TT32
P9		_TT33
P10		_TT34
P11		_TT35
P12		_TT36
P13		_TT37
P14		_TT38
P15		_TT39

SYSTEM 300

CPU Ports

Port Label	Peripheral Code	WMCS Devicename
P0	________________	_TT0
P1	________________	_TT1
P2	________________	_TT2
P3	________________	_TT3

HYDRA STUDENT MONITORS

Port Label	Peripheral Code	WMCS Devicename
P0	Hydra	_HD0
P1	Hydra	_HD1
P2	Hydra	_HD2
P3	Hydra	_HD3
P4	Hydra	_HD4
P5	Hydra	_HD5
P6	Hydra	_HD6
P7	Hydra	_HD7
P8	Hydra	_HD8
P9	Hydra	_HD9
P10	Hydra	_HD10
P11	Hydra	_HD11
P12	Hydra	_HD12
P13	Hydra	_HD13
P14	Hydra	_HD14
P15	Hydra	_HD15
P16	Hydra	_HD16
P17	Hydra	_HD17
P18	Hydra	_HD18
P19	Hydra	_HD19
P20	Hydra	_HD20
P21	Hydra	_HD21
P22	Hydra	_HD22
P23	Hydra	_HD23
P24	Hydra	_HD24
P25	Hydra	_HD25
P26	Hydra	_HD26
P27	Hydra	_HD27
P28	Hydra	_HD28
P29	Hydra	_HD29

A.3 SYSTEM LOG/DIAGNOSTICS

______________________________ ______________________________

user reporting the diagnostic date and time

SYMPTOMS (check all that apply)

____ Stack dump (fill in the diagnostic display on the back of this form).

____ Diagnostic message with no stack dump. Describe: ____________

____

____ Your terminal was still alive following the diagnostic.

____ Your terminal was dead following the diagnostic.

____ Your terminal was the only terminal affected.

____ Other terminals were dead following the diagnostic.

____ The system was still alive following the diagnostic.

____ The system was dead following the diagnostic.

____ Describe any system abnormality not included in the foregoing list:

____

CONTEXT

Describe what you were typing or executing when the incident occurred. List other processes that were running, recently installed hardware or software, and any other items pertinent to the diagnostic:

____

____

____

REPLICATION

Can the incident be reproduced? ____________ If yes, explain how:

____

____

RECOVERY

____ The system was rebooted.

____ RECOVER was executed.

____ Serial port was dismounted.

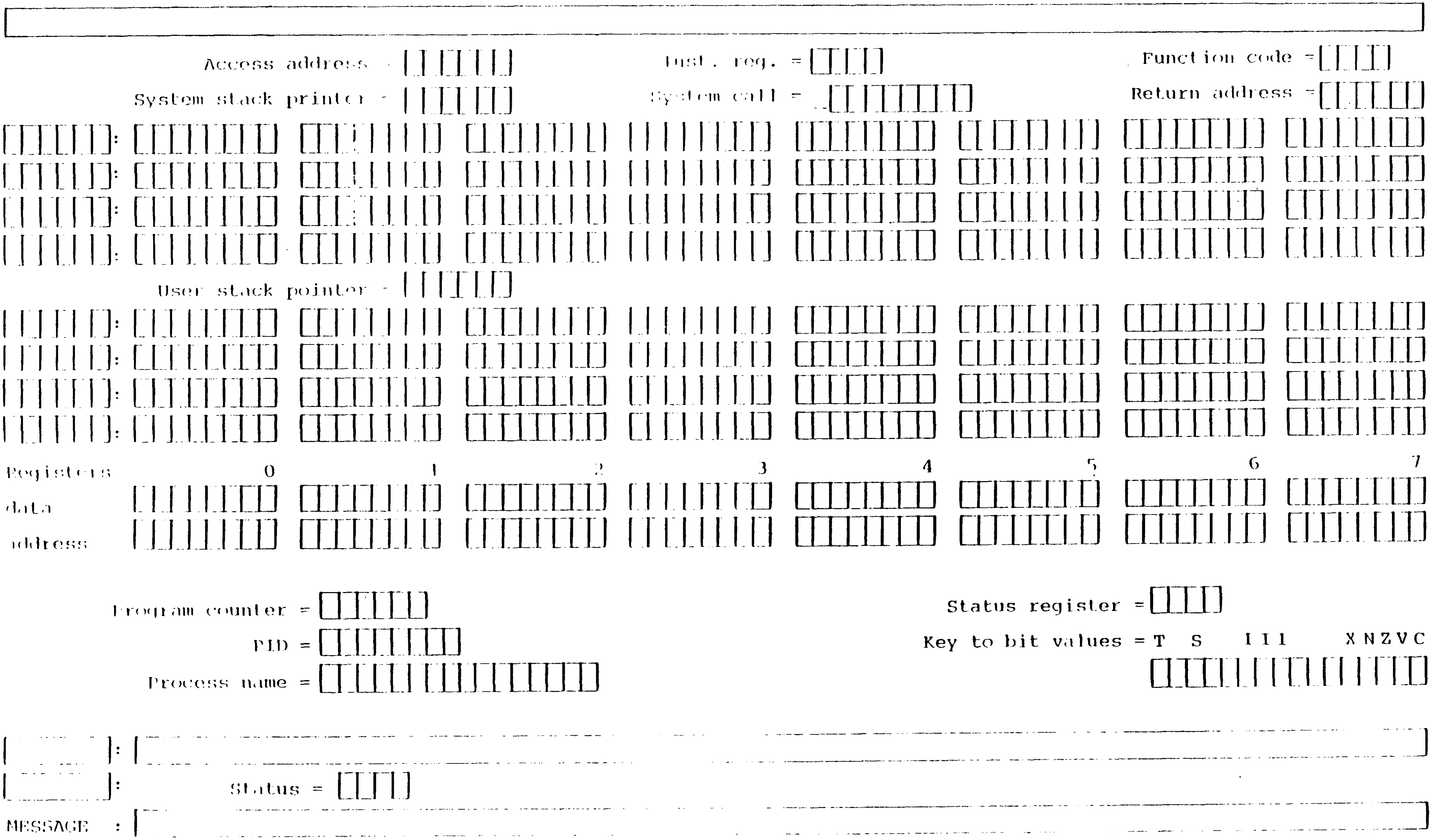
____ A backup file copy was restored.

____ The disk was reinitialized.

____ Hardware was replaced.

____ Other (describe) __________

____



Access address =

Inst. reg. =

Function code =

System stack printer =

System call =

Return address =

User stack pointer =

Registers 0 1 2 3 4 5 6 7

data

address

Program counter =

PID =

Process name =

Status register =

Key to bit values = T S III XNZVC

Status =

MESSAGE :

APPENDIX B

BOOT LOG MESSAGES

B.1 ADEI TAPE CONTROLLER

When a device error occurs on the ADEI tape controller during boot, this kind of message appears:

```
Drive startup failure.   Status = 00xx
```

The "xx" number contains the status register as returned by the hardware. The values of the bits (LSB = 0) in the status register are:

0	The tape is write-enabled or no cartridge is in the drive
1	The drive has done an autorewind
2	EOT has been detected
3	BOT has been detected
4	'ON' with cartridge loaded
5	The drive is rewinding
6	A file mark was detected
7	Reserved for future use

Two error conditions will cause the "Status = 00xx" message to appear: the EOT bit is set, or the File Mark Detected bit is set. If one of these bits is set, other bits will probably also be set.

B.2 CIPHER OR CTC TAPE CONTROLLERS

When a device error occurs on the Cipher or CTC tape controllers during boot, this kind of message appears:

```
Drive startup failure.   Status = xxxx
```

The "xxxx" number actually contains two separate fields in the position "yyzz". The "yy" part of the number is the hardware status register at the time of the error. The "zz" part of the number is the hardware error register at the time of the error. Each of these registers contain bit fields with the following definitions:

The values of the bits (LSB = 0) in the status register are:

Status register

0	The tape is ready
1	An interrupt is pending
2	The drive is rewinding
3	BOT has been detected
4	EOT has been detected
5	ID burst has been detected
6	A file mark was detected
7	The tape is busy

Error register

0	The tape is write protected
1	A controller memory parity error occurred
2	Illegal sequence of commands
3	Illegal command (Cipher only)
4	Illegal on-line transition (Cipher only)
5	Corrected parity error is on the tape
6	Uncorrectable parity error is on the tape
7	An error occurred

One error condition will cause the "Status = xxxx" message to appear: the Error Occurred bit in the error byte is set. If it is set, only one of the other error condition bits in the error byte should be set. The status byte will reflect the status of the controller at the time the error occurred.

B.3 FLNT CONTROLLER

When a device error occurs on the FLNT controller during boot, this kind of message appears:

```
Drive startup failure.   Status = 00xx
```

The "xx" in the message is an error number. The following chart explains each error number:

81	Illegal unit number (drive number)
82	Illegal channel number (drive type)
83	Illegal command
84	Illegal cylinder number
85	Illegal sector number
86	Illegal sector size
87	Illegal head number

88	Drive not initialized
89	No track 0 detected on restore
8a	1795 was busy during drive selection
8b	1795 was busy during seek
8c	Undefined
8d	Undefined
8e	Undefined
8f	Undefined
90	Read error, early or no interrupt during read
91	Write error, early or no interrupt during write
92	Write protected disk error
93	Illegal pre-comp value error
94	Write fault error from 1795
95	Data error during verify
96	Format error (no 1795 interrupt)
97	Record not found error
98	Record not found CRC error
99	Data CRC error
9a	Lost data error
9b	Drive no ready error
9c	Drive not responding...(1795 hung)
9d	Bus error accessing system RAM
9e	Unable to access system bus
9f	Address error, illegal user buffer address
a0	SCSI port already busy on select
a1	Selected SCSI controller not responding
a2	No SCSI request after select
a3	SCSI controller in wrong phase
a4	SCSI controller parity error
a5	SCSI port parity error
a6	Error while requesting SCSI error status
a7	Illegal number of sectors per track
a8	Illegal number of heads (floppy)
a9	SCSI hand-shake entered wrong phase
aa	SCSI port hardware error
ab-bf	Undefined

The following errors are from the SCSI controller on channel 3:

c0	SCSI error with no error status
c1	No index signal
c2	No seek complete
c3	Write fault
c4	Drive not ready
c5	Drive not selected
c6	No track zero
c7	Multiple Winchester drives selected
c8-cc	Undefined
cd	Seek in progress
ce-cf	Undefined

d0	ID read error. ECC error in the ID field
d1	Uncorrectable data error during a read
d2	ID address mark not found
d3	Data address mark not found
d4	Record not found. Found correct cylinder and head but not sector
d5	Seek error. Read/write head positioned on a wrong cylinder and/or selected a wrong head
d6	Undefined
d7	Write protected
d8	Correctable data field error
d9	Bad block found
da	Format error
db	Undefined
dc	Unable to read the alternate track address
dd-de	Undefined
df	Sequencer time out during disk transfer
e0	Invalid command received from the host
e1	Illegal disk address. Address is beyond the maximum address
e2	Illegal function for the current drive type
e3	Volume overflow
e4-fe	Undefined
ff	SCSI controller RAM error

B.4 SCSI CONTROLLER

When a device error occurs on the SCSI controller during boot, the following kind of message appears:

```
Drive startup failure.   Status = 00xx
```

The "xx" in the message is an error number. The following chart explains each error number:

01	No index signal
02	No seek complete (approximately 1 second)
03	Write fault
04	Drive not ready
05	Drive not selected
06	No track 00
07	Multiple Winchester drives selected
08-0c	Undefined
0d	Seek in progress
0e-0f	Undefined
10	ID read error. ECC error in the ID field
11	Uncorrectable data error during a read
12	ID address mark not found
13	Data address mark not found

14 Record not found. Found correct cylinder and head but not sector

15 Seek error. Read/write head positioned on a wrong cylinder and/or selected a wrong head

16 Undefined

17 Write protected

18 Correctable data field error

19 Bad block found

1a Format error. The controller detected that during the Check Track command, the format on the drive was not expected

1b Undefined

1c Unable to read the Alternate Track address. The address of the alternate track cannot be read correctly with no ECC error.

1d-1e Undefined

1f Sequencer time out during disk transfer

20 Invalid command received from the host

21 Illegal disk address. Address is beyond the maximum address

22 Illegal function for the current drive type

23 Volume overflow. Maximum sector address was passed during a multiple sector read or write

24-2f Undefined

30 Controller RAM error

B.5 SMD CONTROLLER

When a device error occurs on the SMD controller during boot, the following kind of message appears:

```
Drive startup failure.   Status = 00xx
```

The "xx" in the message is an error number. The following chart explains each error number:

10 Disk not ready

11 Invalid disk address

12 Seek error

13 ECC code error—data field

14 Invalid command code

15 Invalid track in IOPB

16 Invalid sector in command

17 Undefined

18 Bus time out

19 Write error

1a Disk write protected

1b Unit not selected

1c No address mark—header field

1d No data mark—data field

1e Unit fault

1f Data overrun time out

20 Surface overrun

21 ID field error—wrong sector read

22	ID field ECC error
23	Uncorrectable error
24	Undefined
25	Undefined
26	No sector pulse
27	Data overrun
28	No index pulse on write format
29	Sector not found
2a	ID field error—wrong head
2b	Invalid sync in data field
2c	Invalid sync in header field
2d	Seek time out error
2e	Busy time out
2f	No normal complete at beginning of a seek
30	RTZ timeout
31	Format overrun on data
32	Undefined
33	Undefined
34	Undefined
35	Undefined
36	Undefined
37	Undefined
38	Undefined
39	Undefined
3a	Undefined
3b	Undefined
3c	Undefined
3d	Undefined
3e	Undefined
3f	Undefined
40	Unit not initialized
41	Disk busy executing
42	Gap specification error
43	ANSI bus timeout—type 1
44	ANSI bus timeout—type 2
45	ANSI bus timeout—type 3
46	ANSI bus error
47	Illegal command
48	Illegal parameter
49	Time dependent command error
4a	Common reject
4b	Seek error
4c	Mapped header error
4d	Unspecified seek error
4e	Read/write fault
4f	Disk formatting error
50	Sector/track specification error
51	Byte/sector specification error
52	Interleave specification error
53	Invalid head address
54	Undefined
55	Invalid head address

B.6 WINCHESTER/FLOPPY CONTROLLER

When a device error occurs on the Winchester/floppy controller the following kind of message appears:

Drive startup failure. Status = xxxx

The "xxxx" number actually contains two separate fields in the position "yyzz". The "yy" part of the number is the hardware status register at the time of the error. The "zz" part of the number is the hardware error register at the time of the error. Each of these registers contain bit fields with the following definitions: (LSB = 0)

Status register	
0	An error occurred
1	The device is write protected
2	An interrupt is pending
3	A bus error occurred on the data transfer
4	The seek operation completed
5	A write fault occurred
6	The drive is ready for the next command
7	The drive is busy

Error register	
0	The Data Address Mark was not found
1	Track 0 was not found
2	The command aborted
3	An internal consistency error occurred
4	The ID field was not found
5	A CRC error occurred in the ID field
6	A CRC error occurred in the data field
7	A bad block was detected

For example, suppose the error message is "Status = 5140". In this case the status register is 51. This means that the drive is now ready, the seek completed, and that an error occurred.

The error register is 40. This is bit 6 set, meaning that a CRC error occurred in the data field. The most probable explanation for this error is that the data in the sector to be read has gone bad.

Only one error bit should be set in this register at any given time, so the only values that should be seen are 01, 02, 04, 08, 10, 20, 40, and 80. If any other numbers appear, serious problems in the hardware have occurred.

APPENDIX C

SAMPLE COMMAND FILE FOR BUILDING A DISKETTE BOOT VOLUME

```
!====================================================================
!MCSMAINT:  Create an MCS 5.0 maintenance boot diskette for mapped
!                                                 system 150/155/160
!Format:    @mcsmaint
!Procedure: @mcsmaint
!====================================================================
prompt "Insert diskette -- Press [RETURN] when ready"
dinit _dx0 WMCS_5.0_Boot :format
mnt _dx0
!====================================================================
! Build /ROOTDIR/
!====================================================================
pstat 0 :owner=system
option :prot=s:r,o:rw,g:r,p:
copy sys$disk/rootdir/bootdisk.156  _dx0/rootdir/
copy sys$disk/rootdir/devconfig.156 _dx0/rootdir/
copy sys$disk/rootdir/disk.156      _dx0/rootdir/
copy sys$disk/rootdir/kernel.156    _dx0/rootdir/
copy sys$disk/rootdir/osinit.156    _dx0/rootdir/
copy sys$disk/rootdir/startup.156   _dx0/rootdir/
copy sys$disk/rootdir/bootconf.156  _dx0/rootdir/sysconfig.*
copy sys$disk/rootdir/tty.156       _dx0/rootdir/
!====================================================================
! Build /SYSLIB/
!====================================================================
option :prot=s:r,o:rw,g:r,p:r
crd _dx0/syslib/ :prot=s:re,o:rwe,g:re,p:re
copy sys$disk/syslib/bootstart.com         _dx0/syslib/startup
copy sys$disk/syslib/setup25[2-5].sys      _dx0/syslib/
copy sys$disk/syslib/shortuaf.dat          _dx0/syslib/ :prot=s:,g:,p:
```

SAMPLE COMMAND FILE FOR BUILDING A DISKETTE BOOT VOLUME

```
!=====================================================================
! Build /SYSDSR/
!=====================================================================
crd _dx0/sysdsr/
copy sys$disk/sysdsr/tty$156.dsr    _dx0/sysdsr/
copy sys$disk/sysdsr/mdsk$156.dsr   _dx0/sysdsr/
copy sys$disk/sysdsr/null.dsr       _dx0/sysdsr/
copy sys$disk/sysdsr/wd3$156.dsr    _dx0/sysdsr/
copy sys$disk/sysdsr/wd2$156.dsr    _dx0/sysdsr/
copy sys$disk/sysdsr/wdmf$156.dsr   _dx0/sysdsr/
copy sys$disk/sysdsr/disk.cfg       _dx0/sysdsr/
!=====================================================================
! Build /SYSEXE/
!=====================================================================
option :prot=s:re,o:rwe,g:re,p:re
crd _dx0/sysexe/
copy sys$disk/sysexe/cip.exe                  _dx0/sysexe/
copy sys$disk/sysexe/copy.exe                 _dx0/sysexe/
copy sys$disk/sysexe/del.exe                  _dx0/sysexe/
copy sys$disk/sysexe/dir.exe                  _dx0/sysexe/
copy sys$disk/sysexe/dmnt.exe                 _dx0/sysexe/
copy sys$disk/sysexe/dstat.exe                _dx0/sysexe/
copy sys$disk/sysexe/logflush.exe             _dx0/sysexe/
copy sys$disk/sysexe/mnt.exe                  _dx0/sysexe/
copy sys$disk/sysexe/recover.exe              _dx0/sysexe/
copy sys$disk/sysexe/zap.exe                  _dx0/sysexe/
!=====================================================================
! Configure the diskette
!=====================================================================
prompt "Press [RETURN] to configure the maintenance diskette"
config _dx0
dmnt _dx0 :auto
prompt :noresp "Please remove and label the maintenance diskette"
```

APPENDIX D

SAMPLE COMMAND FILE FOR BUILDING A TAPE BOOT VOLUME

```
!==========================================================================
!MCSTAPE:    Create an MCS 5.0 boot tape for mapped system 100/200/220/300
!Format:     @mcstape
!Procedure: @mcstape
!==========================================================================
prompt "Type the tape device name (ct0 or mt0) :" :logical=devname
ifct0 :=   "ct0:=\" \";mt0:=\"!\""
ifmt0 :=   "ct0:=\"!\";mt0:=\" \""
reset :=   "ct0:=     ;mt0:=     "

ifct0
    'devname' devtype := tape
    'devname' driver := adei$100
    'devname' verif := :noverify
    'devname' bs := 4096

ifmt0
    'devname' devtype := tape
    'devname' driver := ciph$100
    'devname' verif := :verify
    'devname' bs := 1024

reset
prompt "Load tape -- Press [RETURN] when ready"
dinit 'devname' WMCS_5.0_Boot :blocksize='bs'
mnt 'devname'
!==========================================================================
! These files are required for booting. They must appear in this order.
!==========================================================================
pstat 0 :owner=system
option :prot=s:re,o:rwe,g:re,p:re
copy sys$disk/rootdir/boottape.100  'devname'/rootdir/
copy sys$disk/rootdir/tapconfig.100 'devname'/rootdir/
copy sys$disk/rootdir/kernel.100    'devname'/rootdir/
copy sys$disk/rootdir/disk.100      'devname'/rootdir/
copy sys$disk/rootdir/tty.100       'devname'/rootdir/
copy sys$disk/rootdir/tape.100      'devname'/rootdir/
```

SAMPLE COMMAND FILE FOR BUILDING A TAPE BOOT VOLUME

```
copy sys$disk/rootdir/osinit.100      'devname'/rootdir/
copy sys$disk/sysdsr/'driver'.dsr     'devname'/sysdsr/
copy sys$disk/sysdsr/tty$100.dsr      'devname'/sysdsr/
copy sys$disk/sysdsr/null.dsr         'devname'/sysdsr/
copy sys$disk/rootdir/startup.100     'devname'/rootdir/
copy sys$disk/sysexe/cip.exe          'devname'/sysexe/
!==================================================================
! end of required files
!==================================================================
!
! The following files are for setting up the memory disk and copying
! files to it.  For example:
!
!    mnt _md0
!    copy _mt0/*/ _md0/*/ :builddir
!
!==================================================================
copy sys$disk/sysexe/mnt.exe          'devname'/sysexe/
copy sys$disk/rootdir/devconfig.100   'devname'/rootdir/
copy sys$disk/sysdsr/mdsk$100.dsr     'devname'/sysdsr/
copy sys$disk/sysexe/copy.exe         'devname'/sysexe/
!==================================================================
! The following files are typical commands that might be on a
! boot tape.  These files should be copied to the memory disk.
! The purpose of these files is to have the tools necessary to
! bring up the primary system disk.
!==================================================================
copy sys$disk/sysexe/cip.exe          'devname'/sysexe/
copy sys$disk/sysexe/copy.exe         'devname'/sysexe/
copy sys$disk/sysexe/del.exe          'devname'/sysexe/
copy sys$disk/sysexe/dinit.exe        'devname'/sysexe/
copy sys$disk/sysexe/dir.exe          'devname'/sysexe/
copy sys$disk/sysexe/dmnt.exe         'devname'/sysexe/
copy sys$disk/sysexe/dstat.exe        'devname'/sysexe/
copy sys$disk/sysexe/logflush.exe     'devname'/sysexe/
copy sys$disk/sysexe/mnt.exe          'devname'/sysexe/
copy sys$disk/sysexe/recover.exe      'devname'/sysexe/
copy sys$disk/sysexe/zap.exe          'devname'/sysexe/
!==================================================================
!  Device driver files typically needed on a boot tape
!==================================================================
copy sys$disk/sysdsr/tty$100.dsr      'devname'/sysdsr/
copy sys$disk/sysdsr/null.dsr         'devname'/sysdsr/
copy sys$disk/sysdsr/disk.cfg         'devname'/sysdsr/
copy sys$disk/sysdsr/'driver'.dsr     'devname'/sysdsr/
copy sys$disk/sysdsr/smd$100.dsr      'devname'/sysdsr/
copy sys$disk/sysdsr/imi$100.dsr      'devname'/sysdsr/
```

```
!=================================================================
!   syslib files typically needed on a boot tape
!=================================================================
copy sys$disk/syslib/bootstart.com  'devname'/syslib/
copy sys$disk/syslib/setup25[2-5].sys 'devname'/sysexe/
!=================================================================
!   Boot tape complete
!=================================================================
dmnt 'devname' :auto
```

APPENDIX E

SAMPLE COMMAND FILES FOR BACKUPS

```
!=============================================================
! Daily incremental backup of sys$disk
!
! This command file edits the backup parameter file, allowing the
! operator to inspect and modify its contents.  Then it mounts the
! tape, skips to the end (so that this daily backup is appended to
! the end of the previous data on the tape), and executes the backup
! command.  Finally, it prints the log file, dismounts the tape,
! purges extra files, and informs the operator that the backup is
! complete.
!=============================================================
vew daily.prm                             ! Edit the parameter file
prompt "Load the tape and press RETURN"  ! Wait till operator is ready
mnt _mt0                                  ! Mount the tape
skip _mt0 :eot                            ! Skip to the end
backup @daily                             ! Perform the backup
print daily.log                           ! Print the log file
dmnt _mt0 :auto                           ! Dismount the tape
pu daily.log,dbackup.prm :auto            ! Purge extra files
send sys$output "Daily backup complete."
!=============================================================
! Daily backup complete.
!=============================================================
```

PARAMETER FILE DAILY.PRM

```
_mt0/20APR84/                     ! Backup directory (today's date)
_ds0/*/*.*                        ! Which files to backup
:mod                              ! Use modification date
:volume=3                         ! Current volume number
                                  ! Dinit command line for tape.
                                  ! Volume label is Monday's date with
                                  ! the relative volume number appended
:init="dinit _mt0 16/APR/84_## :blocksize=4096"
:since=yesterday_07:00            ! All files since yesterday
:log=daily.log                    ! Name the log file

!=============================================================
! Weekly incremental backup of _ds0
```

```
!
! This command file edits the backup parameter file, allowing the
! operator to inspect and modify its contents.  It then executes
! the backup command.  Finally, it prints the log file, dismounts
! the tape, purges extra files, and informs the operator that the
! backup is complete.
!============================================================
vew weekly.prm                           ! Edit the parameter file
prompt "Load the tape and press RETURN"  ! Wait till operator is ready
backup @weekly                           ! Perform the backup
print weekly.log                         ! Print the log file
dmnt _mt0 :auto                          ! Dismount the tape
pu weekly.log,wbackup.prm :auto          ! Purge extra files
send 'sys$output' "Weekly backup is complete"
!============================================================
! weekly backup complete.
!============================================================
```

PARAMETER FILE WEEKLY.PRM

```
_mt0/16APR84/                  ! Backup directory (today's date)
_ds0/*/*.*                     ! Which files to backup
:mod                           ! Use modification date
:preinit                       ! Initialize the tape before starting
                               ! Dinit command line for tape
                               ! Volume label is Monday's date with
                               ! the relative volume number appended
:init="dinit _mt0 16/APR/84_## :blocksize=4096"
:since=09/APR/84_07:00         ! All files since last monday
:log=weekly.log                ! Name the log file

!============================================================
! MONTHLY BACKUP
!
! This command file edits the backup parameter file, allowing the
! operator to inspect and modify its contents.  Then it executes
! the backup command.  Finally, it prints the log file, dismounts
! the tape, purges extra files, and informs the operator that the
! backup is complete.
!============================================================
vew monthly.prm                          ! Edit the parameter file
prompt "Load the tape and press RETURN"  ! Wait till operator is ready
backup @monthly                          ! Perform the backup
print monthly.log                        ! Print the log file
dmnt _mt0 :auto                          ! Dismount the tape
pu monthly*.log,mbackup.prm :auto        ! Purge extra files
send sys$prompt "Monthly backup complete."
!============================================================
! MONTHLY BACKUP COMPLETE.
!============================================================
```

PARAMETER FILE MONTHLY.PRM

```
_mt0/02APR84/                     ! Backup directory (today's date)
_ds0/*/*.*                        ! Which files to backup
:preinit                          ! Initialize the tape before starting
                                  ! Dinit command line for tape
                                  ! Volume label is today's date with
                                  ! the relative volume number appended
:init="dinit _mt0 02/APR/84_## :blocksize=4096"
:logfile=monthly.log              ! Name the log file
```

APPENDIX F

SAMPLE SYSTEM COMMAND FILES

```
!=================================================================
! STARTUP.COM is executed automatically during the system boot.
!
! WICAT Systems maintains this file (i.e., it is updated automatically
! when a new or revised product—released and supported by
! WICAT—is installed).
!
! Do not modify this file.
!=================================================================
OPTION :NOLOG
INSTALL @SYS$DISK/SYSLIB/INSTALL.PRM ! Install privileged images
@SYS$DISK/SYSLIB/DEVICEUP.COM     ! Mount and set status on all devices
@SYS$DISK/SYSLIB/APPLICUP.COM     ! Bring up all languages and applications
@SYS$DISK/SYSLIB/LOCALUP.COM      ! Site specific commands
OPTION :LOG
&LOGFLUSH
SEND * "'sys$sysname' is up and running."
LOG :NOLOG

!=================================================================
! DEVICEUP.COM (for Systems 150, 155, 160)
!
! This command file is executed automatically during the system boot, and
! does three things:
!
!   1. Makes logical name assignments pertaining to devices.
!
!   2. Installs device drivers.
!
!   3. Mounts devices and assigns attributes to mounted devices.
!
! If you access this file as instructed in step 5 (page 6-4) of chapter 6
! in the WMCS Introductory System Manager Manual, use [CTRL] p to page
! forward in this file until you come to the heading:
!
!    MOUNT DEVICES AND ASSIGN CHARACTERISTICS TO MOUNTED DEVICES
!
```

```
! Read and follow the instructions under that heading.  When you
! complete those instructions, perform step 6 (also on page 6-4) of
! chapter 6 in the WMCS Introductory System Manager Manual and continue
! with the steps in that manual.
!
! The commands used in this command file are described in the WMCS User
! Reference Manual.  See the WMCS System Manager Reference Manual for an
! explanation of ownership and protection.
!
! The system manager maintains this file and must update it whenever
! necessary (DEVICEUP.COM is not modified automatically whenever a new
! or revised version of a product is installed).  Read the WMCS System
! Manager Reference Manual for information on maintaining DEVICEUP.COM.
!=====================================================================
! MAKE LOGICAL NAME ASSIGNMENTS PERTAINING TO DEVICES.
!=====================================================================
sys$pipe   :=== "sys$disk/sysdsr/pip02$156.dsr"
sys$print  :=== _pq0
!=====================================================================
! INSTALL DEVICE DRIVERS.
!=====================================================================
install 'sys$pipe'
cd sys$disk/sysdsr/
install adei$156.dsr,\
        cent$156.dsr,\
        ciph$156.dsr,\
        mdsk$156.dsr,\
        que$156.dsr,\
        rvd3$156.dsr,\
        smd$156.dsr,\
        tty$156.dsr,\
        wd2$156.dsr,\
        wd3$156.dsr,\
        wdmf$156.dsr
!=====================================================================
! MOUNT DEVICES AND ASSIGN CHARACTERISTICS TO MOUNTED DEVICES
!
! INSTRUCTIONS: When your system is booted, the WMCS goes through this
! section to find out what ports on your system are to be mounted and
! what attributes are to be assigned to each port.
!
! The following lines appear in this file for each port that can be
! mounted on a System 150, 155, or 160:
!
! ! Port: p1  Username/function:
!        mnt   _tt1
!        dstat _tt1 :termtype=t7000 :expandtabs :owner=[0,1] :default
!
! The WMCS ignores the first line because the line begins with an
! exclamation mark. However, you can fill in the name of the user (or the
! function for which you are using that port) for your reference.
!
```

```
! The second line tells the WMCS to mount the port and name it _TT1.
!
! The  third  line  tells  the  WMCS to assign various attributes to port
! _TT1.  Terminal type is the first attribute specified.
!
! Based  on  the  list  of  ports  you  completed  on  p. 6-3 of the WMCS
! Introductory   System   Manager   Manual,   go   through  the following
! subsections  in  this  file  and  delete  the  exclamation  mark at the
! beginning of the second and third lines of any device you want mounted:
!
!    MOUNT (AND ASSIGN ATTRIBUTES TO) PORTS BELONGING TO THE FIRST I/O
!      (OR ICI) BOARD
!
!    MOUNT (AND ASSIGN ATTRIBUTES TO) PORTS BELONGINT TO THE SECOND I/O
!      (OR ICI) BOARD
!
! When you have deleted the  exclamation mark  at the  beginning of those
! command lines  that  will mount  the ports on your system, use the list
! you filled out on p. 6-3 to make sure  the value appearing in this file
! next to :TERMTYPE=  for  each  mounted port matches the peripheral code
! you wrote down for that port when you filled out the list.
!
! When  the  value  next  to :TERMTYPE= is correct for each mounted port,
! complete the instructions under  the following subsection in this file:
!
!    MOUNT (AND ASSIGN ATTRIBUTES TO) THE PORT(S) TO WHICH YOUR
!    PRINTER(S) IS(ARE) CONNECTED
!
!========================================================================
!  Mount (and assign attributes to) ports belonging to the first I/O
!    (or ICI) board
!========================================================================
! Port: p0  Username/function:
!   mnt _tt0     !This port is mounted automatically at boot time
!   dstat _tt0 :termtype=t7000 :expandtabs :owner=[0,1] :default
! Port: p1  Username/function:
!   mnt _tt1
!   dstat _tt1 :termtype=t7000 :expandtabs :owner=[0,1] :default
! Port: p2  Username/function:
!   mnt _tt2
!   dstat _tt2 :termtype=t7000 :expandtabs :owner=[0,1] :default
! Port: p3  Username/function:
!   mnt _tt3
!   dstat _tt3 :termtype=t7000 :expandtabs :owner=[0,1] :default
! Port: p4  Username/function:
!   mnt _tt4
!   dstat _tt4 :termtype=t7000 :expandtabs :owner=[0,1] :default
! Port: p5  Username/function:
!   mnt _tt5
!   dstat _tt5 :termtype=t7000 :expandtabs :owner=[0,1] :default
! Port: p6  Username/function:
!   mnt _tt6
!   dstat _tt6 :termtype=t7000 :expandtabs :owner=[0,1] :default
! Port: p7  Username/function:
!   mnt _tt7
!   dstat _tt7 :termtype=t7000 :expandtabs :owner=[0,1] :default
!==========================================================================
!  Mount (and assign attributes to) ports belonging to the second I/O
!    (or ICI) board.
!==========================================================================
! Port: p8  Username/function:
!   mnt _tt8
!   dstat _tt8 :termtype=t7000 :expandtabs :owner=[0,1] :default
! Port: p9  Username/function:
!   mnt _tt9
!   dstat _tt9 :termtype=t7000 :expandtabs :owner=[0,1] :default
! Port: p10 Username/function:
!    mnt _tt10
!    dstat _tt10 :termtype=t7000 :expandtabs :owner=[0,1] :default
! Port: p11 Username/function:
!    mnt _tt11
!    dstat _tt11 :termtype=t7000 :expandtabs :owner=[0,1] :default
! Port: p12 Username/function:
!    mnt _tt12
!    dstat _tt12 :termtype=t7000 :expandtabs :owner=[0,1] :default
! Port: p13 Username/function:
!    mnt _tt13
!    dstat _tt13 :termtype=t7000 :expandtabs :owner=[0,1] :default
! Port: p14 Username/function:
!    mnt _tt14
!    dstat _tt14 :termtype=t7000 :expandtabs :owner=[0,1] :default
! Port: p15 Username/function:
!    mnt _tt15
!    dstat _tt15 :termtype=t7000 :expandtabs :owner=[0,1] :default
!==========================================================================
! Mouunt (and assign attributes to) the  port(s) to which your printer(s)
!  is(are) connected.
!
!  INSTRUCTIONS: Four  command-line  character  strings  (similar to the
!  following)  must  appear  in  this  section  for  each printer on your
!  system.
!
!       mnt _tt1 :special
!       dstat _tt1 :protection=s:rwe,p:,g:,o:rwed :owner=[1,1] :default
!       mnt _pq0
!       dstat _pq0 :assocdev=_tt1 :quemgrres :owner=[1,1] :default\
!        :printtype="qprint _pq0 :nonewline"
!
!  The first  line  mounts  port _TT1 as a special device (i.e., a device
!  that LOGFLUSH does not check).  If your printer  is connected to port
!  2 or 3,  etc.,  you  would  substitute  _TT2,  _TT3,  etc.,  for _TT1.
!
!  The  second  line   assigns   attributes   to  port  _TT1.   Read  the
!  description  of  DSTAT  in the WMCS User Reference Manual for detail.
```

```
!
! NOTE: Do not change the protection or ownership specified.
!
! The third line mounts the print queue handler, _PQ0 (PQ signifying
! print queue and 0 indicating that this particular print queue is the
! first one on the system—if you mount another print queue you would
! specify _PQl, etc.).
!
! The fourth command-line character string assigns attributes to _PQ0.
! Note that for the :ASSOCDEV= Switch the port you specify is the port
! to which the printer is connected. For information about the value
! (in double quotation marks) assigned to :PRINTTYPE=, read the
! description of the QPRINT Command in the WMCS User Reference Manual.
!
! Note that this example uses port _ttl. This port is mounted here
! and are commented out above. If your printer is attached to another
! port, delete the exclamation marks above that correspond to _ttl and
! place an exclamation mark in front of the lines that mount the port
! to which your printer is attached.
!
! When you complete these instructions, return to p. 6-4 of the WMCS
! Introductory System Manager Reference Manual and complete the steps
! in that manual.
!==================================================================
! Print queue 0
!       mnt _ttl :special
!       dstat _ttl :protection=s:rwe,p:,g:,o:rwed :owner=[1,l] :default
!       mnt _pq0
!       dstat _pq0 :assocdev=_ttl :quemgrres :owner=[1,l] :default\
!                  :printtype="qprint _pq0 :nonewline"
! Print queue 1
        mnt _tt2 :special
        dstat _tt2 :protection=s:rwe,p:,g:,o:rwed :owner=[1,l] :default
        mnt _pql
!       dstat _pql :assocdev=_tt2 :quemgrres :owner=[1,l] :default\
!                  :printtype="qprint _pql :nonewline"
!==================================================================
!  Mount the parallel port(s) on your system
!==================================================================
! Mount 1st parallel port
!    mnt _pp0 :special
!    dstat _pp0 :owner=[0,l] :default
! Mount 2nd parallel port
!    mnt _ppl :special
!    dstat _ppl :owner=[0,l] :default
!==================================================================
! Mount other hard-disk drives on your system
!==================================================================
! Additional SMD disk drives (SMD474, SMD168, SMD84)
!     mnt _dsl
!     mnt _ds2
!     mnt _ds3
! Additional Winchester disk drives (WIN12, WIN19, WIN30, WIN43)
!     mnt _dc1
!     mnt _dc2
!     mnt _dc3

!=================================================================
! LOCALUP.COM is executed automatically during the system boot.  Use this
! file for logical  name assignments, etc., that you want executed during
! the system boot.
!
! The   system   manager   maintains   this   file  (it  is  not  modified
! automatically when a new or revised product is installed).
!=================================================================
!===============================================================
! System logical name assignments
!===============================================================
!d*ir       :== "dir"
ds          :== "dir :size :create :username :head"
home        :== "cd \'sys$home\'"
kill        :== "pstat :kill"
lnk         :== "link :symbol=_HEAP"
me          :== "stat :uic=\'sys$username\'"
!pr*int     :== "print"
st*at :== "pstat :head :systemstatus :port :status :size :prior :scheduled :time
sys$sysname:== "System name"
sys$tmp     :== 'sys$disk'
!ti*me      :== "time"
!ty*pe      :== "type"
!===============================================================
! Other local system commands
!===============================================================
del sys$tmp/systmp/*.*.* :auto
time :prompt

!=================================================================
! APPLICUP.COM is executed automatically  during  the system boot so that
! the  software products on your system (such as UltraCalc, PASCAL, etc.)
! are ready for use.  For example,  if UltraCalc  is one of the software
! products on your system,  the  following kind of command-line character
! string, preceded by an at-sign, in this file  would  set up the logical
! name assignments, define the  necessary terminal characteristics, etc.,
! so that when the system is  booted, all a  user  needs  to  do  to  use
! UltraCalc  is type UC  and strike [RETRN]:
!
!      sys$disk/ultclc/ultup.com
!
! APPLICUP.COM is updated automatically (i.e., the command-line character
! string  for a software product  is inserted in this file automatically)
! as part of the  installation  of  any  product  that  is  released  and
! supported by WICAT Systems.
!=================================================================
!=================================================================
! LOGON.COM    is   executed   each   time   a   user   logs   on to the system.
!
! WICAT Systems maintains this file  (i.e.,  it is  updated automatically
! when    a    new    or    revised    product—released    and    supported  by
! WICAT Systems—is installed).
!
! Do not modify this file.
!=================================================================
@SYS$DISK/SYSLIB/LOCALON
@USERUP

!=================================================================
! LOCALON.COM  is executed each  time  a user logs on to the system.  Use
! this  file  for  logical  name  assignments,  etc., that pertain to all
! users.
!
! The system  manager maintains   this  file (i.e., it  is  not  modified
! automatically when a new or revised product is installed).
!=================================================================
logline sys$disk/syslib/syslog.dat "'sys$username' logged on 'sys$input'"
type sys$disk/syslib/lognews.txt :pause
option :prompt="'sys$username'> " :perm

!=================================================================
! USERUP.COM — This  file is  executed  each time a  user logs on to the
! system.  Each  user  should  have  a  userup.com  file  in  his  "home"
! directory.  In  this file  the user can place commands that he wants to
! execute  each time he logs on.  The  userup.com in /syslib/ is reserved
! for the system manager's account.
!=================================================================
time

!=================================================================
! LOGOFF.COM   is   executed   each   time   a   user   logs off of the system.
!
! WICAT Systems maintains this file  (i.e.,  it is  updated automatically
! when    a    new    or    revised    product—released    and    supported  by
! WICAT Systems—is installed).
!
! Do not modify this file.
!=================================================================
cd 'sys$home' :nolog
@sys$disk/syslib/localoff
@'sys$home'useroff

!=================================================================
! LOCALOFF.COM  is executed each time a user logs off of the system.  Use
! this file for any actions to be taken when users log off of the system.
!
! The system  manager maintains   this  file (i.e., it  is  not  modified
! automatically when a new or revised product is installed).
```

```
!=================================================================
logline sys$disk/syslib/syslog.dat "'sys$username' logged off 'sys$input'"

!=================================================================
! USEROFF.COM  is executed each time a user logs off of the system.  Each
! user has an instance of this file in his or her home directory.
!
! Each user maintains this file in his or her home directory.
!=================================================================
```

WICAT Systems, Inc.

Product-documentation Comment Form

We are constantly improving our documentation, and we welcome specific comments on this manual.

Document Title: ______________________________

Part Number: ______________________________

Your Position:

□ Novice user

□ Experienced user

□ Applications programmer

□ System manager

□ Systems analyst

□ Hardware technician

Questions and Comments **Page No.**

Briefly describe examples, illustrations, or information that you think should be added to this manual.

______________________________ __________

______________________________ __________

______________________________ __________

What would you delete from the manual and why?

______________________________ __________

______________________________ __________

______________________________ __________

What areas need greater emphasis?

______________________________ __________

______________________________ __________

______________________________ __________

List any terms or symbols used incorrectly.

______________________________ __________

______________________________ __________

______________________________ __________

First Fold

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 00028 OREM, UTAH

POSTAGE WILL BE PAID BE ADDRESSEE

WICAT Systems, Inc.
Attn: Corporate Communications
1875 S. State St.
Orem, UT 84058

Second Fold

Tape