CDC® VSOS VERSION 2

FOR USE WITH
CYBER 200 SERIES
COMPUTER SYSTEM

Installation Handbook

CONTROL DATA

60459440

TOOL INDEX

This index lists each VSOS tool and the number of the page on which it is described.

CDC® VSOS VERSION 2

FOR USE WITH
CYBER 200 SERIES
COMPUTER SYSTEM

Installation Handbook

60459440

REVISION RECORD

REVISION	DESCRIPTION
A (04-16-82)	Manual released.
B (10-15-82)	Revised to reflect VSOS 2.0 corrective code release at PSR level 575. This edition obsoletes all previous editions.
C (07-29-83)	Revised to reflect VSOS 2.1 at PSR level 592 and to make literary and technical changes. Because extensive changes are made, change bars and dots are not used and all pages reflect the latest revision level. This edition obsoletes all previous editions.
D (03-30-84)	Revised to reflect VSOS release 2.1.5 at PSR level 607 and to make literary and technical changes. Because extensive changes are made, change bars and dots are not used and all pages reflect the latest revision level. This edition obsoletes all previous editions.
E (10-31-85)	Revised to reflect VSOS release 2.2 at PSR level 644 and to make literary and technical changes. Because extensive changes are made, change bars and dots are not used and all pages reflect the latest revision level. This edition obsoletes all previous editions.
F (04-18-86)	Revised to reflect VSOS release 2.2.5 at PSR level 654 and to make literary and technical changes. Because extensive changes have been made, change bars and dots are not used and all pages reflect the latest revision level. This edition obsoletes all previous editions.
G (12-05-86)	Revised to reflect VSOS release 2.3 at PSR level 670 and to make literary and technical changes. Because extensive changes have been made, change bars and dots are not used and all pages reflect the latest revision level. This edition obsoletes all previous editions.
H (10-23-87)	Revised to reflect VSOS release 2.3.5 at PSR level 690 and to make literary and technical changes. A major change is the addition of chapter 7, which describes security safeguards. Technical changes are indicated by bars in the margin. A dot by the page number indicates the entire page is changed. A bar by the page number indicates only the pagination is changed.
J (11-15-88)	Revised to reflect VSOS release 2.3.7 at PSR level 712 and to make literary and technical changes. Technical changes are indicated by bars in the margin. A dot by the page number indicates the entire page is changed. A bar by the page number indicates only the pagination is changed.

Publication No.
60459440

REVISION LETTERS I, O, Q, S, X AND Z ARE NOT USED.

Address comments concerning this manual to:

Control Data
Technical Publications ARH219
4201 North Lexington Avenue
St. Paul, Minnesota 55126-6198

or use Comment Sheet in the back of this manual.

LIST OF EFFECTIVE PAGES

New features, as well as changes, deletions, and additions to information in this manual, are indicated by bars in the margins or by a dot near the page number if the entire page is affected. A bar by the page number indicates pagination rather than content has changed.

PAGE	REV
Front Cover/ Inside Front Cover	-
Title Page	-
2	J
3/4	J
5	J
6	G
7	J
8	J
9	J
1-1	J
1-2	H
2-1	J
2-2	H
2-3	H
2-4	J
2-5	J
2-6	H
2-7	J
2-8	J
2-9	J
2-10	G
2-11	G
2-12	G
2-13	G
2-14	J
3-1	J
3-2	J
3-3	J
3-4	J
4-1	G
4-2	G
4-3	H
4-4	J
4-5	J
4-6	J
4-7	J
4-8	J
4-9	J
4-10	J
5-1	J
5-2	J
5-3	J
5-4	G
5-4.1/5-4.2	J
5-5	G
5-6	G
5-7	J
5-8	J
5-8.1	J
5-8.2	J
5-9	J
5-10	J
5-11	J
5-12	J
5-12.1/5-12.2	J
5-13	J
5-14	J
5-14.1/5-14.2	J
5-15	J
5-16	H
5-17	J
5-18	H
5-19	G
5-20	J
5-21	J
6-1	J
6-2	H
6-3	H
6-4	H
6-4.1	J
6-4.2	H
6-5	H
6-6	H
6-7	G
6-8	G
6-9	G
6-10	H
6-11	H
6-12	G
6-13	G
6-14	G
6-15	G
6-16	G
6-17	G
6-18	G
6-19	G
6-20	G
6-21	G
6-22	J
6-23	G
6-24	H
6-25	J
6-26	J
6-27	J
6-28	G
6-29	G
6-30	J
6-31	G
6-32	G
6-33	G
6-34	J
6-35	G
6-36	G
6-37	H
6-38	J
6-39	G
6-40	G
6-41	G
6-42	G
6-43	H
6-44	H
6-45	H
6-46	H
6-47	H
6-48	H
6-49	H
6-50	J
6-51	H
6-52	G
6-53	G
6-54	G
6-55	G
6-56	G
6-57	G
6-58	G
6-59	J
6-60	G
6-61	H
6-62	H
6-63	J
7-1	H
7-2	H
7-3	H
7-4	H
7-5	H
7-6	H
7-7	H
7-8	H
7-9	H
7-10	H
7-11	H
A-1	J
A-2	J
A-3	J
A-4	G
A-5	G
B-1	H
Index-1	J
Index-2	J
Index-3	H
Comment Sheet	J
Back Cover	-

PREFACE

This handbook describes procedures for installation and maintenance of the CONTROL DATA® Virtual Storage Operating System (VSOS) Version 2.3.7 on the CDC® CYBER 200 Model 205 Computer System. It includes procedures to follow for a first-time installation, as well as procedures for system upgrades from VSOS 2.3 and 2.3.5 to VSOS 2.3.7.

AUDIENCE

This handbook is intended to be used by analysts who have experience working with VSOS.

Analysts should be familiar with the VSOS operator commands described in the VSOS 2 Operator's Guide.

ORGANIZATION

This handbook is divided into these chapters:

- Chapter 1 provides an introduction to the installation process and a description of the materials used.
- Chapter 2 describes an initial installation.
- Chapter 3 describes procedures for upgrading from a 2.3 or 2.3.5 system to a 2.3.7 system.
- Chapter 4 gives information on maintaining the system.
- Chapter 5 describes the installation parameters.
- Chapter 6 describes tools and utilities useful for building and maintaining the system.
- Chapter 7 describes the establishment and management of a security sensitive site system.
- Appendix A describes additional installation and configuration procedures involving the Remote Host Facility (RHF).
- Appendix B describes additional installation notes including the use of SCHEDULE.

CONVENTIONS

In this manual, values preceded by the # character are hexadecimal.

RELATED PUBLICATIONS

Refer to the following documents for additional information on CYBER 200 hardware and software.

Control Data Publication	Publication Number
CYBER 200 Assembler Version 2 Reference Manual	60485010
CYBER 200 Maintenance Software System Reference Manual	60457200
CYBER 200 Model 205 Computer System Hardware Reference Manual	60256020
CYBER 200 Model 205 Troubleshooting Guide	60430060
CYBER 200 Software Standards and Conventions	17329020
FORTRAN 200 Version 1 Reference Manual	60480200
Remote Host Facility Usage	60460620
Software Peripheral Drivers Reference Manual	96769390
VSOS 2 Operator's Guide	60459430
VSOS 2 Reference Manual, volume 1	60459410
VSOS 2 Reference Manual, volume 2	60459420
VSOS Site Manager's Handbook	60461490
VSOS User's Guide for FORTRAN 200 Programmers	60455390
65252-1 Network Access Device Hardware Reference Manual	60458510

Control Data manuals can be ordered from:

Literature and Distribution Services
STP005
308 North Dale Street
St. Paul, Minnesota 55103

DISCLAIMER

This product is intended for use only as described in this document. Control Data Corporation cannot be responsible for the proper functioning of undescribed features or parameters.

Control Data no longer supports the FORTRAN 66 compiler.

CONTENTS

APPENDIXES

INDEX

FIGURES

TABLES

OVERVIEW 1

This chapter outlines procedures for installing a VSOS 2.3.7 system on the CYBER 205.

INSTALLATION OVERVIEW

NEW SITES

- Gather materials. See Preparing for Installation later in this chapter.
- Complete the initial installation procedures described in chapter 2.
- Maintain the system using the procedures described in chapter 4, as required.

SITES UPGRADING FROM VSOS 2.2.5 OR 2.3 TO 2.3.5

- Gather materials. See Preparing for Installation later in this chapter.
- Perform upgrade procedures described in chapter 3.
- Maintain the system using the procedures described in chapter 4, as required.

PREPARING FOR INSTALLATION

Before you begin either an initial installation or a system upgrade, follow the procedures described next.

RELEASE DOCUMENTATION

A software release bulletin (SRB) is issued with each VSOS release. It contains last minute information not included in this manual. It also contains a list of the installation materials.

Installation bulletins (IBs) are issued at any time after release when new information or code becomes available which would affect users of the release.

In addition to using the Installation Handbook, you will need to refer to the VSOS Operator's Guide; the VSOS Reference Manual, volumes 1 and 2; and the CYBER 200 Maintenance Software System Reference Manual. These manuals and their publication numbers are listed in the preface.

RELEASE MATERIALS

Materials that are released with VSOS, including disk packs, tapes, assembly listings, and the binaries of the test mode utilities, are listed in the software release bulletin (SRB) that accompanies the VSOS release. Read the SRB, check the release materials against the list in the SRB, and gather all release materials.

NOTE

Several VSOS system files are included on the Maintenance Control Unit (MCU) system pack A in units 3, 4, 5, and 6. When you have completed the installation and have backed up the system files, you can reinitialize units 5 and 6 on SMD pack A to make space available for site use.

INITIAL INSTALLATION 2

INTRODUCTION

Follow the procedures described in this chapter to initially install VSOS on a CYBER 205. This initial system uses only one million words of memory and is used as a running system for building the user's system.

The release materials include:

- The Maintenance Control Unit (MCU) software and skeleton VSOS system (contained on SMD pack A).
- The full VSOS system release (contained on VSOS tape SL712 or SMD packs A, B, and C/D).
- The 2.3.7 program libraries (PLs) (contained on VSOS tape SL712 or SMD pack C/D).
- The FORTRAN system release (contained on VSOS tape SL712 or SMD pack C/D).

INITIALIZING MCU HARDWARE AND SOFTWARE

Follow this procedure to initialize MCU hardware and software:

1. Mount pack A on unit 0.
2. Autoload the MCU. Refer to the VSOS 2 Operator's Guide for detailed information.
3. Examine and change the site-dependent parameters of the MCU. (Refer to the PARM command in the CYBER 200 Maintenance Software System Reference Manual if you need further information.) Use the PARM command to display site-dependent parameters. Use the ALTER command to change site-dependent parameters.

INITIALIZING THE SYSTEM CONFIGURATION TABLE (SCT)

The following command sequence initializes the system configuration table (SCT) with hardware configuration information. (Disregard the tape commands if your site does not have CYBER 200 tapes.)

```
SYST
NAD,mid,MID,port
NAD,dcd,DCD,port,ch,access
NAD,dad,DAD
NAD,tcd,TCD,port,ch,access
NAD,tad,TAD
NAD,rcd,RCD,port,,access
CONR,rcd,snad
COND,dcd,dad
CONT,tcd,tad
DISK,dev,dad,unit,,,status,use
TAPE,dev,tad1,tad2,pu,status,use,rdonly
EXIT
```

mid	Maintenance interface device (MID) number.
port	Port number (1 through #10) of the CYBER 205 I/O ports to which the device is connected.
dcd	Disk channel device (DCD) number.
tcd	Tape channel device (TCD) number.
ch	Channel expander (SCEX) number (1 through 4); 0 for SCA.
access	Access code.
dad	Disk access device (DAD) number.
tad	Tape access device (TAD) number.
rcd	Remote channel device (RCD) number.
snad	NAD number and trunk control unit (TCU). Refer to the VSOS 2 Operator's Guide if you need further information.
dev	Device number.
unit	Physical unit number.
status	UP (default)/DOWN.
use	ON (default)/OFF.
pu	Physical unit number of the tape unit on the controller (0 through 7).
rdonly	RW (tape unit can read and write data)/RO (unit can read only).

CREATING A PATH FROM MCU TO DISK

NOTE

Before beginning initial installation procedures, ensure that the central processing unit (CPU) and all network access devices (NADs) are stopped. Also ensure that all disk packs are switched off-line, except for the one used for initial installation. The NADs must be manually cleared by pressing the Master Clear button because the TMC command clears only NADs that are configured.

The following command sequence creates a path from the MCU to the disk by loading controlware into the appropriate NADs.

```
LNAD
LNAD,dad
LNAD,dcd,,,2
```

dad Disk access device (DAD) NAD number.

dcd Disk channel device (DCD) NAD number.

Load the DAD NAD before loading the matching DCD NAD.

INITIALIZING THE DISK

Use the following procedure to create a label, a permanent file index (PFI), and a directory of file segmentation (DFS) on the disk used for initial installation.

1. Read the bad sector map from the disk into central memory and print it out. The disk used for initial installation must have a valid bad sector map. Refer to Organization of Disk Packs (under NAMEPACK Disk Labeling Utility) in chapter 6 for more information.

   ```
   LCEN,dev,0,11F82,1
   DUMP,0,8000
   ```

   dev Device number.

2. To change the label and permanent file index,

   a. Read the initial label and pack file index file (LABELPFI) into central memory by entering:

      ```
      LCM,LABELPFI,12
      ```

   b. Verify that none of the files defined in LABELPFI conflicts with the flaws printed out in step 1. Refer to the LABELPFI initial structure shown in figure 2-1. (Refer to the pack file index entry (file index table) in volume 2 of the VSOS 2 Reference Manual for general information on the format of PFI entries.) You must change the location of all conflicting files in the copy of LABELPFI in central memory to eliminate the conflicts. Files Q5VSF, CKPTFILE, CONFIG1, RSLOD, and VSLOD must remain nonsegmented files.

   c. To make the size of the PFI on the disk smaller than the released value (#3D blocks), change byte 2 of word B of the entry for the file PFIxx to the new length of the PFI. Also, change the rightmost 32 bits of word 9 (highest bit address = length of PFI * #8000 + 1) to reflect the new highest bit address. In the label portion of LABELPFI, change word 4 to reflect the new PFI length.

   d. After changes are made, write the file LABELPFI back onto the SMD pack by entering:

      ```
      SCM,LABELPFI,12
      ```

3. Write the initial label and pack file index and system files onto the disk using one of the following commands:

```
JOB INIT819,12,,dn,ds
```

dn Two-digit device number.

ds Two-digit device set number.

Figure 2-1 shows the initial LABELPFI structure.

File Name	Memory Location (Bit Address)	Begin Sector	End Sector	Length
CETR91	8000	0	22F	230
LABEL	8400	230	230	1
PFI	8800	231	26D	3D
CONFIG1	8C00	26E	26F	10
OPERATOR	9000	27E	3ED	170
BATCHPRO	9400	3EE	43D	50
Q5AUTOS	9800	43E	441	4
READMCU	9C00	442	481	40
RSLOD	A000	482	4C1	40
SWITCH	A400	4C2	501	40
Q5VSF	A800	502	E01	900
VSLOD	AC00	E02	FC1	1C0
IQM	B000	FC2	1041	80
SHRLIB	B400	1042	12A1	260
Q5POOLST	B800	12A2	12B9	12
DFS	BC00	12C2	12E1	20
DEAD01	C000	12E2	1AE1	800
CKPTFILE	C400	1AE2	24E9	A08
BADS81	C800	11F82	11F82	1
CETR92	CC00	11F94	121AF	21C
CETR93	D000	24144	2421C	168

Figure 2-1. Initial LABELPFI Structure

INITIALIZING VSOS

Follow this procedure to autoload and initialize VSOS:

1. Enter the autoload command:

   ```
   AUTO
   ```

2. Enter the date and time in the format requested.

3. Select the initial configuration by entering a 1 in response to the prompt by AUTOCON.

4. Initialize the AUTOCON parameters. (A detailed description of these parameters can be found in the VSOS 2 Operator's Guide.) The parameters that are normally changed are the RHF application names (APPL commands), the RHF physical/logical identifiers (RHFID commands), the default output lid (OLID), and the device set parameters (DVST).

5. After making all desired changes to the AUTOCON parameters, enter:

   ```
   SAVE,1
   GO
   ```

6. Wait for the system-loaded message from VSOS.

LOGGING IN AS INTERACTIVE USER (ON THE MCU)

Enter the following command sequence to log in as an interactive user from the MCU.

```
TTY
(Change display to ROLL mode)
LOGIN 999998
```

ESTABLISHING FILES FOR THE 2.3.7 SYSTEM

1. Execute the following sequence:

```
READMCU,I=INIT,ID=12
SWITCH,INIT,RT=R
BATCHPRO,INIT / TL=1000
```

2. Perform the set of procedures, listed next, that corresponds to your 2.3.7 release materials media (VSOS tapes or SMD packs).

   - Installation from VSOS tape:

     a. Mount tape SL712.

     b. Enter:

```
LOADPF / TL=1000 / PO=SL712,IL712,DEV=NT,VSN=SL712,SEL=R,TF=LB,DS=dvst.
```

        dvst Device set identifier. Six characters in the format DVSTxx, where xx is the device set number.

NOTE

Pool SL712 contains the 2.3.7 running system built for a one-million-word machine, and pool IL712 contains the 2.3.7 program libraries (PLs and static controllees, LOAD, SLGEN, and IMPL) which can be used as the input pool for rebuilding the system.

- Installation from SMD packs:

  a. Mount release pack B, then submit the following job:

```
MOVEB,STlid.
USER,U=999998,AC=404SOV,PA=SYSTEM.
RESOURCE,TL=999.
READMCU,I=MOVEB,O=TEMP,ID=12.
SWITCH,TEMP,RT=R,T=P.
BEGIN,,TEMP,pack.
```

  lid Logical identifier of the CYBER 205.

  pack Pack identifier. Six characters in the format PACKxx, where xx is the pack number in ASCII notation.

  b. Mount release pack C/D, then submit the following job:

```
MOVECD,STlid.
USER,U=999998,AC=404SOV,PA=SYSTEM.
RESOURCE,TL=999.
READMCU,I=MOVECD,O=TEMP,ID=12.
SWITCH,TEMP,RT=R,T=P.
BEGIN,,TEMP,pack.
```

  lid Logical identifier of the CYBER 205.

  pack Pack identifier. Six characters in the format PACKxx, where xx is the pack number in ASCII notation.

NOTE

Packs B and C/D are to be mounted on MCU drive 1.

VERIFYING INITIAL INSTALLATION

The following procedure verifies that the initial installation is completed and operational:

```
PATTACH,SL712
COPY,INSTAT,TEMP
SWITCH,TEMP,INSTAT
STATUS
```

The output file (OUTPUT) contains information about all public files. In addition, OUTPUT will display an identification list from each of the program libraries (PLs).

To save this file for later inspection, switch OUTPUT to a logical file name (lfn) of your choice.

INSTALLING ON-LINE DIAGNOSTICS AND JSEAS

Autoload the new system in pool SL712.

Install the on-line diagnostics as described in appendix D of the CYBER 200 Maintenance Software System Reference Manual.

Install the job-submitted error analysis system (JSEAS) as described in section 7 of the CYBER 200 Maintenance Software System Reference Manual.

NOTE

When you autoload the installed system, ensure that the AUTOCON parameters VIRT, RESI, SYSID, and SLIB are set for use with pool SL712.

DISK PACK FORMAT

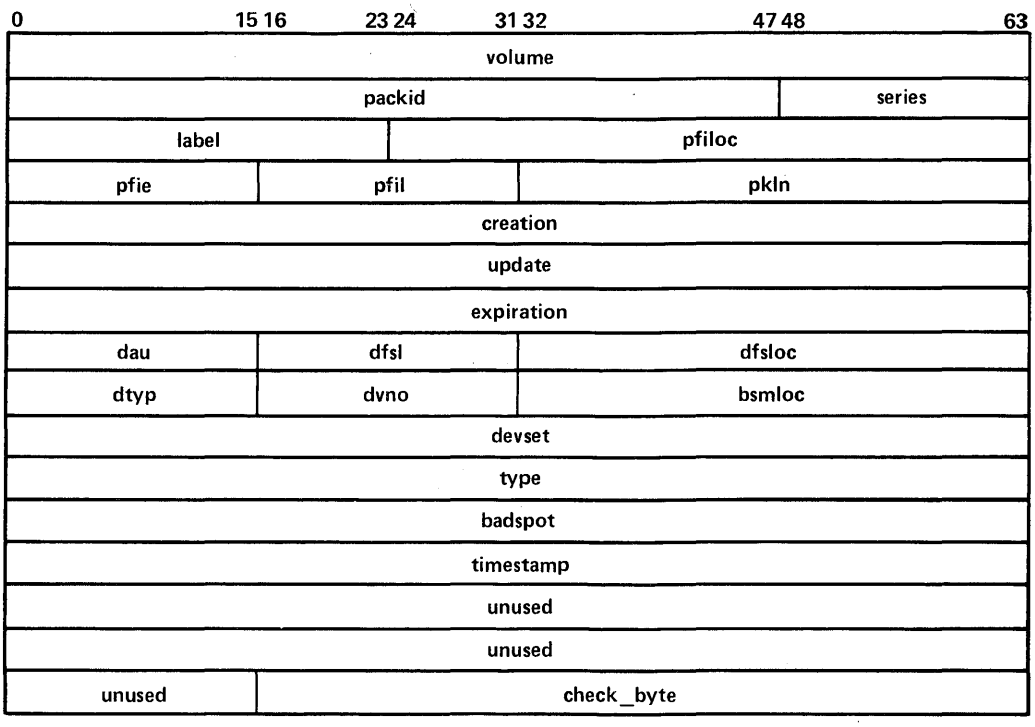
CE DIAGNOSTIC AREAS

These areas are reserved on the disk pack for CE diagnostics.

PACK LABEL FORMAT

The pack label on the disk pack provides specific information about the disk pack (see figure 2-2).

The system expects to find the label at block #230 unless something is physically wrong with that block. If the pack label is not at block #230, the system tries to locate it up to eight more times, each time adding #F0, before logically shutting down the disk pack.



Word	Name	Bits	Description
0	volume	0-63	The pack label volume name VOL3, left-justified and blank-filled.
1	packid	0-47	The pack identifier, in the format PACKxx, where xx is the pack number in ASCII notation.

Figure 2-2. Pack Label Format (Sheet 1 of 3)

Word	Name	Bits	Description
1	series	48-63	The pack series number; the default is 2031 in hexadecimal notation.
2	label	0-23	Disk block address of the label.
	pfiloc	24-63	Disk block address of the first block of the PFI.
3	pfie	0-15	Entry number of LABEL entry in the PFI.
	pfil	16-31	The length of the PFI in blocks.
	pkln	32-63	The pack length, which is the number of 512-word blocks that can be allocated on this disk pack.
4	creation	0-63	ASCII date of the creation of this label in the format mm.dd.yy.
5	update	0-63	ASCII date of the creation of this label in the format mm.dd.yy.
6	expiration	0-63	ASCII date of the creation of this label in the format mm.dd.yy.
7	dau	0-15	The binary number of 512-word blocks in a disk allocation unit. This is the allocation unit for this disk pack.
	dfsl	16-31	The length of the directory of file segmentation (DFS) in blocks.
	dfsloc	32-63	The starting disk block address of the DFS.
8	dtyp	0-15	The device type: 3 81922 disk pack (double-density, 18 sector).
	dvno	16-31	The device number associated with this pack.
	bsmloc	32-63	The starting disk block address of the bad spot map (BSM).
9	devset	0-63	The device set name in the format DVSTxx, left-justified and blank-filled, where xx is the device set number.
10	type	0-63	The type of disk pack; 81922 in hexadecimal notation (only for compatibility with release version 2.1.6).

Figure 2-2. Pack Label Format (Sheet 2 of 3)

Word	Name	Bits	Description
11	badspot	0-63	The ASCII name of the BSM (only for compatibility with release version 2.1.6).
12	timestamp	0-63	The date and time of the last autoload (used primarily by checkpoint restart).
13	unused	0-63	Unused.
14	unused	0-63	Unused.
15	unused	0-15	Unused.
	check_byte	16-63	The check sum of selected fields of the pack label. Words 1 through 4 and 8 through 12 are used to generate the check_byte.

Figure 2-2. Pack Label Format (Sheet 3 of 3)

PERMANENT FILE INDEX (PFI)

Each pack contains a permanent file index (PFI) with entries of 16 words each. The pack label, the PFI, CE diagnostic areas, the bad spot map, and the DFS appear in the PFI as file entries. All public and private files on the pack require entries in the PFI. Unused entries are filled with the hexadecimal pattern, #000C1F1C000C1F1C.

The location of the first block of the PFI is variable and is indicated in the pack label. The length is variable up to a maximum of #40. The name of the PFI is PFInn, where nn is the pack number of the disk.

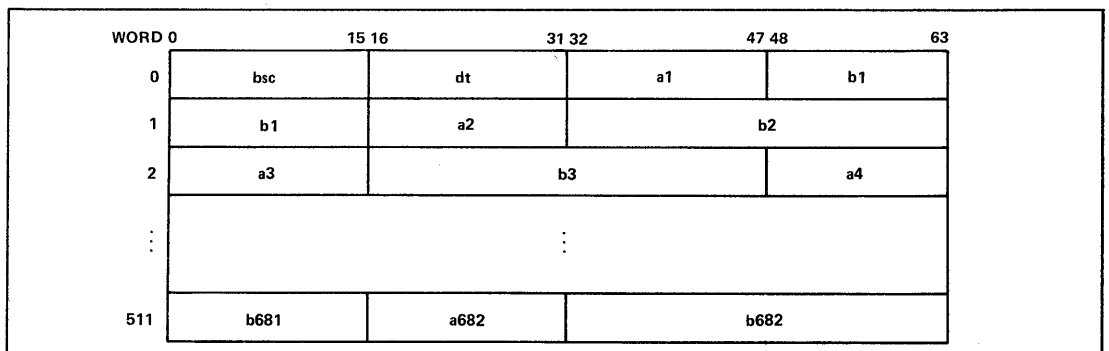
DIRECTORY OF FILE SEGMENTATION (DFS)

File segment information is stored in the directory of file segmentation (DFS) file, which is managed by the system. There is a DFS file on each pack with the name DFSnn, where nn is the pack number. This file is created when the disk is initialized. DFSnn contains segment and allocation information for all VSOS files that have segments on the pack. An entry in the PFI is linked to the first file segment table entry in the DFS for that file. Additional DFS entries are linked together as needed. Note that when a file overflows from one pack to another, DFS entries from each pack are linked together. There is only one PFI entry for the file, and it is located on the pack that contains the first file segment. Each DFS entry is four words long, with the first word of each entry containing a pointer to the next DFS entry if one exists.

BAD SPOT MAP

This map is produced through the use of the NHCD off-line diagnostic. If no bad spots exist, the map area is set to zeros.

The bad spot map typically is in sector #11F82 on double-density, 18-sector 819 disks, but it is pointed to by the BSMLOC field in word 8 of the Pack Label (see Pack Label Format in this chapter). It is a one-block file containing the entries shown in figure 2-3.



Word	Name	Description
0	bsc	Bad spot count, the number of bad spots in the map.
	dt	Device type, #819.
0 to 511	an	Bad spot address tag.
	bn	Bad spot block sector number. The field may be split between two central memory words.

Figure 2-3. Bad Spot Map Format

SYSTEM UPGRADE 3

INTRODUCTION

Follow the procedures described in this chapter to upgrade your system from either VSOS 2.3 or VSOS 2.3.5. Conversion to 2.3.7 from systems prior to 2.3 requires that you first upgrade to 2.3 or 2.3.5.

MOVING 2.3.7 FILES TO DISK

1. Create (or select) a privileged user number to be used for system builds.

2. Create pools SL712 and IL712 owned by the user number from step 1, which allows access by system user number 999998.

To install a running one-million-word 2.3.7 system into pool SL712, proceed with step 3. Otherwise, to install the 2.3.7 program libraries (PLs) and necessary utilities (LOAD, SLGEN, and IMPL) into pool IL712 to build on a running 2.3 or 2.3.5 system, go to step 4.

3. Installation from VSOS tapes:

   a. Log in to system user 999998.

   b. Mount tape: SL712.

   c. Enter:

      ```
      LOADPF / TL=1000 / PO=SL712,IL712,DEV=NT,VSN=SL712,DS=dvst,SEL=R.
      ```

      dvst Six-character device set identifier in the format DVSTxx, where xx is the device set number.

Installation from SMD packs:

a. Mount release pack A, then submit the following job:

```
MOVEA,STlid.
USER,U=user number,AC=account number,PA=password.
READMCU,I=MOVEA,O=TEMP,ID=12.
SWITCH,TEMP,RT=R,T=P.
BEGIN,,TEMP,pack.
```

b. Mount release pack B, then submit the following job:

```
MOVEB,STlid.
USER,U=user number,AC=account number,PA=password.
READMCU,I=MOVEB,O=TEMP,ID=12.
SWITCH,TEMP,RT=R,T=P.
BEGIN,,TEMP,pack.
```

c. Mount release pack C/D, then submit the following job:

```
MOVECD,STlid.
USER,U=user number,AC=account number,PA=password.
READMCU,I=MOVECD,O=TEMP,ID=12.
SWITCH,TEMP,RT=R,T=P.
BEGIN,,TEMP,pack.
```

lid	Logical identifier of the CYBER 205.
user number, account number, and password	The user number, account number, and password that is the owner of pools SL712 and IL712.
pack	Pack identifier. Six characters in the format PACKxx, where xx is the pack number in ASCII notation.

NOTE

Packs A, B, and C/D are to be mounted on MCU drive 1.

4. Installation of PLs and build utilities, LOAD, SLGEN, and IMPL only from VSOS tapes.

   a. Log in to system user 999998.

   b. Mount tape: SL712.

   c. Enter:

      LOADPF / TL=1000 / PO=,IL712,DEV=NT,VSN=SL712,DS=dvst,SEL=R.

dvst	Six-character device set identifier, in the format DVSTxx, where xx is the device number.

   Installation from SMD packs:

   a. Mount release pack C/D, then submit the following job:

      ```
      MOVECD,STLID.
      USER,U=user number,AC=account number,PA=password.
      READMCU,I=MOVECD,O=TEMP,ID=12.
      SWITCH,TEMP,RT=R,T=P.
      BEGIN,,TEMP,pack.
      ```

lid	Logical identifier of the CYBER 205.
user number, account number, and password	The user number, account number, and password that is the owner of pools SL712 and IL712.
pack	Pack identifier. Six characters in the format PACKxx, where xx is the pack number in ASCII notation.

5. Conversion of Q5POOLS to Q5POOLST.

   VSOS 2.3.7 uses a pool list that is twice as long as the VSOS 2.3.5 pool list, requiring that the existing pool list be converted. A new tool, CVPOOL, is provided for this purpose. Refer to chapter 6 for a complete description of CVPOOL.

   To convert the existing file Q5POOLS to the new file Q5POOLST, do the following:

   a. Log in to system user 999998.

   b. Attach pool SL712.

   c. Attach file Q5POOLS.

   d. If permanent file Q5POOLST already exists, purge or rename it.

   e. Enter:

      CVPOOL.

   f. Define the Q5POOLST file that was created.

NOTE

If step 4 is performed instead of step 3, step 5 cannot be performed until the new 2.3.7 system is built.

ESTABLISHING 2.3.7 CONFIGURATION FILES

Follow these steps to autoload the 2.3.7 system:

1. Mount release pack A, and autoload the MCU.

2. If you are upgrading from VSOS 2.3, the format of the system configuration file for the 2.3.7 system is different from that used by the 2.3 system. Consequently, either manually change the site-dependent parameters and build the system configuration table as described in chapter 2 of the VSOS 2 Operator's Guide or use the following automated procedure:

   - Mount the 2.3 system pack A on MCU disk drive 1.
   - Enter the following command to copy the 2.3 configuration file to MCU unit 4:

     ```
     CPYF,SCTFILE,SY,14,SCTFILE,SY,4
     ```

   - Enter the following command to execute a command buffer that converts/ creates a 2.3.7 format configuration file using the 2.3 file as input:

     ```
     JOB,CVTSCAT,SY,4
     ```

   - Enter the following command to copy the new 2.3.7 format configuration file back into SCTFILE:

     ```
     CPYF,NSCTF,SY,4,SCTFILE,SY,4
     ```

   - Reautoload the MCU.

3. Autoload the 2.3.7 system by entering:

   ```
   AUTO
   ```

   Enter the date and time when requested. When the configuration file is requested, enter:

   ```
   CONFIG=CONFIG1,SL712
   ```

   Change the autoload parameters, as necessary. The autoload parameters of the released system are described in chapter 2 of the VSOS 2 Operator's Guide. Parameters RESIDENT, VIRTUAL, SYSID, and SLIB must be changed to use the system in pool SL712. Other parameters that must be changed are the RHF application names, RHF physical/logical identifiers, and the device set parameter (DVST). After making the changes, enter:

   ```
   SAVE,1,CONFIG1,SL712
   GO
   ```

   If the autoload sequence stops, indicating that file Q5IFI is not found, enter CONTINUE, and the system will create it.

MAINTAINING THE SYSTEM 4

FILE NAMING CONVENTIONS

The naming conventions described next apply to files used during the course of system build procedures. These conventions are designed to facilitate the recognition of files and help reduce build errors.

PROGRAM LIBRARIES

Program libraries (PLs) contain source code files and internal information in a special format that only the UPDATE utility can process. (Refer to VSOS 2 Reference Manual, volume 1 for further information.) All program library names terminate with the characters PL. Currently, the following program libraries are available:

BLDPL	System build procedures
F2CPL	FTN200 compiler
F2RPL	FTN200 runtime
HMATHPL	Half-precision library
IMPLPL	IMPL compiler
LIBPL	System library
METAPL	META assembler
RHFPL	RHF utilities and library
SOSPL	Operating system
SPSPL	System utilities
SYSRPL	Static library
TOOLPL	Tools

MODIFICATIONS TO PROGRAM LIBRARIES

All program libraries are modified via a modification file. The modification file includes the details of update processing and any new lines to be added to the program library. To name the modification file, use the name of the program library to be modified, prefixed with the letter I. Thus, ISOSPL contains the modifications for SOSPL, IMETAPL contains the modifications for METAPL, and so forth.

An *IDENT card follows the last set of modifications for each PL to further delineate these sets of modifications. These final identifiers are standardized as follows:

BLDPL	BLDLnnns	METAPL	MTLnnns
F2CPL	F2CLnnns	RHFPL	RHLnnns
F2RPL	F2RLnnns	SOSPL	OSLnnns
HMATHPL	HMLnnns	SPSPL	UTLnnns
IMPLPL	IMLnnns	SYSRPL	SYSLnnns
LIBPL	LBLnnns	TOOLPL	TLLnnns

nnn The three-digit number of the system release level.

s A user-chosen one-character suffix to uniquely identify each system built for a release level.

LIBRARY FILES

Library files are files of modules and a directory of module names and entry points generated by the object library editor (OLE) utility.

Library files terminate with the characters LIB. The following libraries exist on VSOS:

F200LIB	FTN200 runtime library
RHFLIB	RHF libraries
SHRLIB	Dynamic system library
SYSLIB	Static system library
USER1LIB	System utility library
SSYSLIB	Random function library

USING SHARED LIBRARIES (SHRLIB)

DYNAMIC UTILITIES

Dynamic utilities are system utilities that require the system shared library to execute. These utilities will link SIL and FTN200 runtime routines dynamically. The system is released with dynamic utilities that require SHRLIB to be active.

To install a dynamic utility, follow these steps:

1. Compile the utility.
2. If the shared library is active, the loader automatically builds a dynamic controllee. A dynamic utility may also be built when SHRLIB is not active by specifying LINK=D to the loader.
3. Replace the static utility with the dynamic one.

SHARED UTILITIES

A shared utility's code resides in SHRLIB instead of in the utility's controllee file. The controllee file will contain only the minus page, zero page, data bases, and labeled common blocks. Shared utilities require SHRLIB to be active for the utility to be able to execute.

To install a shared utility, follow these steps:

1. Compile the utility.
2. Construct a directive file for SLGEN that contains a directive for adding the utility to SHRLIB. (Refer to the VSOS 2 Reference Manual, volume 1, for further information on the use of SLGEN.)
3. Run SLGEN, specifying the system SHRLIB as the old shared library. SLGEN will process the directive file and construct a new system shared library containing the utility.
4. Replace the old SHRLIB with the new SHRLIB built by SLGEN. Note that an active SHRLIB cannot be replaced. To use the newly-created shared library, enter the AUTOCON SLIB command. (Refer to the VSOS 2 Operator's Guide if you need further information on the use of the SLIB command.)
5. Replace the old utility with the new utility specified with the directive for SLGEN.

STATIC UTILITIES

Static utilities do not require SHRLIB to execute. The static utilities execute SIL and FTN200 runtime routines that were placed into the controllee when it was loaded.

To install a static utility, follow these steps:

1. Compile the utility.
2. Specify LINK=M to the loader.

BUILDING A VSOS 2.3.7 SYSTEM

To build VSOS and its products, follow these steps:

1. Create a privileged user number to be used in the following build steps.

2. Establish a pool owned by the user number created in step 1, which will be used to contain the output of the system build. The recommended pool naming convention is:

   SLnnns

   nnn The three-digit number specifying the system release level.

   s A user-chosen, one-character suffix that uniquely identifies each system built for a release level.

3. Execute the following job to modify the build procedure program library (BLDPL). The system is released to build a shared library containing FTN200 and BATCHPRO. If other options are desired, changes must be included in this job.

```
BLDIN,STlid.
USER,U=user number,AC=account number,PA=password.
RESOURCE,TL=999.
PATTACH,ipool.
PURGE,BLDPLX,PO=ipool.
UPDATE,F,P=BLDPL,N=BLDPLX,C=0.
GIVE,BLDPLX,PO=ipool.
-EOR-
*ID BLDLnnns
```

   ipool The name of a pool that contains the input program library. (This pool must not include controllees for UPDATE, PURGE or GIVE unless they will execute properly on the running system.)

4. Place the modifications to the system into files that have the same name as the program library, prefixed with the letter I (for example, modifications to SOSPL are stored in file ISOSPL).

5. Perform the build steps required for the site (refer to figure 4-1, which shows all of the build steps) by entering parameters into the build job shown next and submitting it for execution for each step.

   The system modifications, input program libraries, and controllees to be used during the build may be local, in the input pool, in the system pool, or in the public file set.

   The output pool cannot be the input pool or the system pool. Refer to the level parameter description that follows.

   The following build job assumes that all required build components, PL's, modifications, and any special controllees, are contained in pool ipool.

```
bldjob,STlid.
USER,U=user number,AC=account number,PA=password.
RESOURCE,TL=9000.
PATTACH,ipool.
IN,bldjob.
UPDATE,Q,D,8,P=BLDPLX,C=BLDTEMP,L=0.
BEGIN,,BLDTEMP,nop,level,linkopt.
```

   ipool — The name of a pool that contains the input program libraries and modification files.

   level — The level identifier to be used for the system being built or modified. This identifier is used as the last four characters of the output pool (SLxxxx) and the version identifier for all controllees built. The recommended format for these four characters is nnns, where nnn is the three-digit number specifying the system release level and is a user-chosen, one-character suffix that uniquely identifies each system built for a release level (refer to step 2).

   linkopt — D Dynamic load.
   M Static load.
   (Refer to Use of Shared Libraries, earlier in this chapter.)

   bldjob — Refer to figure 4-1, which shows how each of the following procedures fits into the system build flow.

XSYSR	XLIB1	XHMT	XFTR	XFTC1	XUTL1
XOPS	XRHF1	XLIB2	XLIB2M	XFTC2	XUTL2
XUSR1	XTOOL	XMETA	XIMPL	XRHF2	XSUTL

   nop — For site use.

NOTE

It is normal for procedure XFTC2 to have two unsatisfied externals, Q9CATMOD and Q9FLINK, on the load of the FTN200 controllee.

Parts of the system may also be built individually by updating the appropriate program library with one of the UPDATE program library (UPD) procedures, then running the procedure with the name of the part that is to be built. If a controllee is being built statically, then the libraries residing on the user where the build is run are used to load the controllee.

Update procedures:

UPDUT	-	Update SPSPL
UPDOS	-	Update SOSPL
UPDTL	-	Update TOOLPL
UPDRHF	-	Update RHFPL

Procedures that build parts of the system on SOSPL:

ANALYZER	AUTOCON	IQM	ITFS	NAMEPACK	OPERATOR
RESIDENT	SCHEDULE	USER1LIB	VIRTUAL	FAULTS	IOSTAT
SYSTAT					

Procedures that build parts of the system on SPSPL:

ATTACH	BATCHPRO	BLANK	CHARGE	COMPARE	COPY
COPYL	DEBUG	DEFINE	DIVERT	DMAP	DROP
DUMP	EDITPUB	EDITUD	EDITUDP	FILES	GIVE
LABEL	LIMITS	LISTAC	LOAD	LOOK	OLE
PACCESS	PASSWORD	PATTACH	PCREATE	PDELETE	PDESTROY
PDETACH	PERMIT	PERMITP	PFILES	PURGE	Q
READMCU	REQUEST	RETURN	REWIND	SKIP	SLGEN
SWITCH	SWITCHP	TASKATT	UPDATE	WRITEMCU	

Procedures that build parts of the system on TOOLPL:

ACC3	ASDF	BADSPOT	BEGIN	BLDIFILE	CHECK
CLEARHG	CLEARMPN	CLEARPMP	CLEARPRF	CTX	CVPOOL
DISKS	EDIT	FILEI	FIND	IN	DELIVER
IOTST	LFNS	MERGE	NADUMP	PAGE	IOQ7LIB
PURGEMCU	READHG	READPMP	READPRF	RELABEL	PULLMOD
SOT	SPY2LIB	SPYR200	STATUS	SUM	SHORTEN
SYSCOPY	XREF	ZAP			

Procedures that build parts of the system on RHFPL:

AUDIT	DUMPF	IRF	LOADPF	MFLINK	MFQUEUE
PTFS	QSTATUS	QTF	QTFS	RHFLIB	SUBMIT

6. Follow these steps to install AUTOCON on the MCU pack:

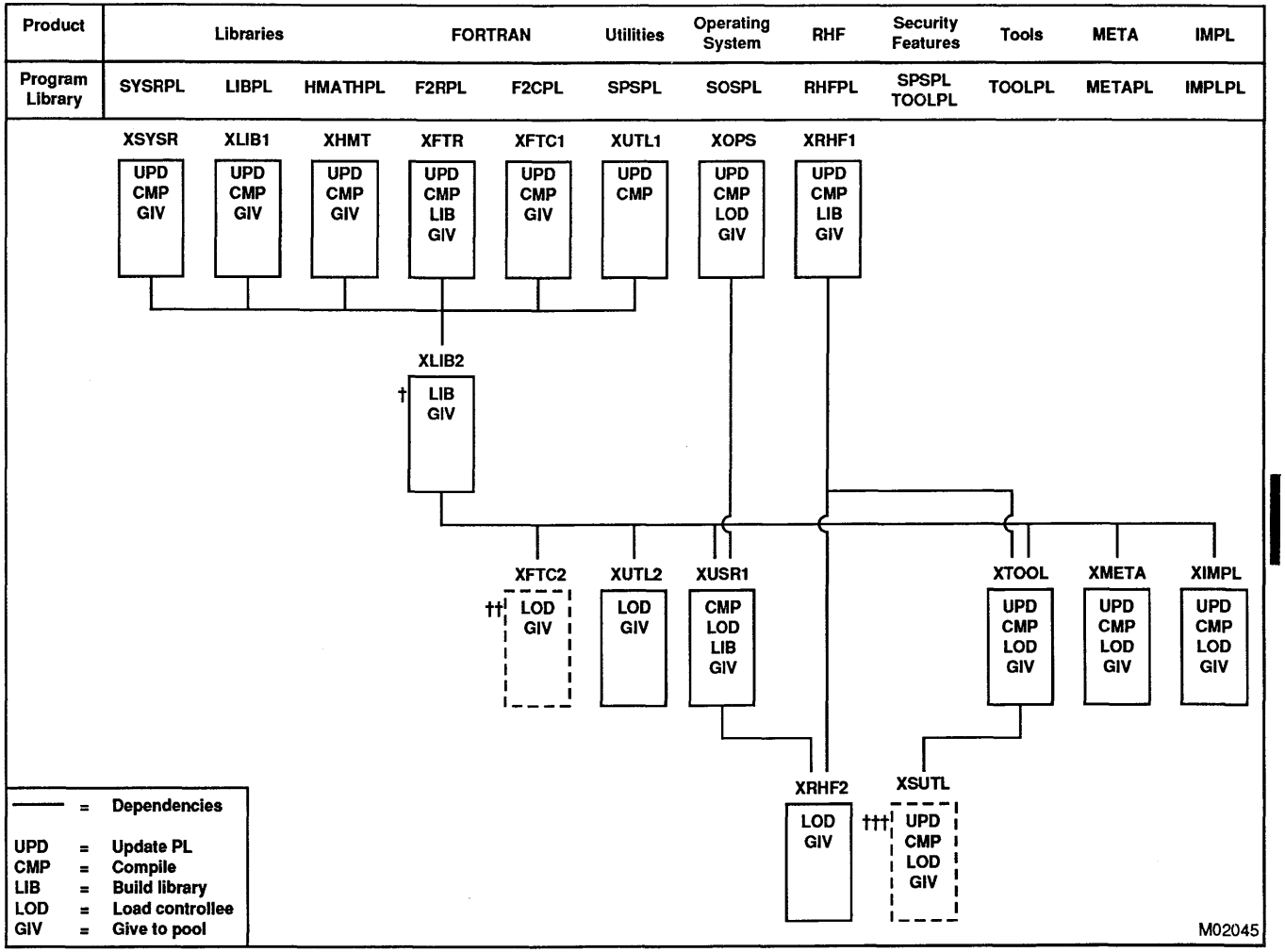
   a. Mount the MCU pack to contain AUTOCON on drive 1.

   b. Log in to a privileged user number.

   c If not already attached, attach the pool that contains the new AUTOCON.

   d. To install the controllee AUTOCON, enter:

```
WRITEMCU,O=AUTOCON,ID=SY,U=14
```



† XLIB2M should be used in place of XLIB2 if the site is building a nonshared library system. All build jobs should be run with the LINKOPT=M parameter.
†† XFTC2 builds a static FTN200 controllee. This job should be run only at sites that do not have FTN200 in their shared library.
††† XSUTL should be run only at security-conscious sites, as described in section 7.

Figure 4-1. Build Flow

INSTALLING L712 FTN200 ON 2.3 OR 2.3.5 SYSTEMS

Follow the procedure for installing IL712, which includes the FTN200 UPDATE Program Libraries HMATHPL, F2CPL, and F2RPL, and the static controllee UPDATE, as described in chapter 3, System Upgrade, in this manual. The L712 UPDATE must be used for this installation because of a change in the L712 Program Library structure. The new UPDATE is compatible with earlier version Program Libraries, and the site may wish to replace their system copy of UPDATE with the L712 version. Once these PLs are installed, they should be used in place of the PLs released under any earlier level. The installation procedure outlined in revisions F or G of the VSOS Version 2 Installation Handbook should be used to reinstall FTN200 and F200LIB. If the shared library (SHRLIB) is utilized, it must also be rebuilt to incorporate the new FTN200 and F200LIB modules. Build jobs XHMT, XFTR, XFTC1, and XLIB2 from the 2.3 or 2.3.5 BLDPL must be rerun to accomplish this. The remainder of the system need not be rebuilt as long as the input files used by these jobs have been retained.

When the compiler is reloaded using the standard installation job, the version number parameter, LEVEL, should be set to Level 712 to properly reflect the product level.

CONFIGURATION CHANGES

To add a new disk pack, refer to NAMEPACK Disk Labeling Utility in chapter 6.

To add network access devices (NADs) and channel expanders, refer to chapter 2, System Autoload and Recovery, in the VSOS 2 Operator's Guide.

CREATING NAD DUMP FILES

Ten NAD dump files are provided with the VSOS 2.3.7 MCU release packs (pack A). Eight of these files are reserved for use by the new NAD dump and reload (NDR) system implemented for use with 2.3 and later systems, the remaining two files are to be used for manual dumps. The NDR system only supports DCD, DAD, and RCD type NADs; therefore, each CYBER 205 system requires both types of NAD dump files.

The dump files used by the NDR system have a fixed nomenclature and length. The file name is Nx where x is a single, hexadecimal digit (0 through F); the ID is SY. The dump file length is 8040 (hexadecimal). The eight NDR files provided with the release system have the file names N0, N1, N2, ..., N7, respectively.

The NDR system can support up to 16 (decimal) dump files. Additional dump files can be created by using the following command:

```
DEFI,Nx,SY,0,8040,unit,NAD DUMP FILE Nx
```

x One of the following digits or characters: 89ABCDEF.

unit The appropriate unit number.

For NAD dumps generated for the NDR system, the NAD dump program generates an ASCII formatted trailer starting at word offset 8000 containing the following information:

```
AUTO NAD DUMP
NAD TYPE: xxx
DATE: XX/XX/XX
TIME: XX:XX
ASSOCIATED NAD:  nn  nn
FAILED NAD: nn
```

The ASSOCIATED NAD field will be blank if the failed NAD is either a DCD or a DAD type NAD.

New NDR system dump files may be created anytime, but the NDR system will not use the newly created dump files until the AUTO or OPER command is entered at the MCU. See the VSOS 2 Operator's Guide for more information concerning the NDR system.

Dump files to be used for manual dumps do not have a fixed nomenclature or length. These dump files should have a length of 8040, since the NAD dump program will generate an ASCII formatted trailer starting at word offset 8000, containing the following information if the file's length is 8040:

```
MANUAL NAD DUMP
NAD TYPE; xxx
DATE: XX/XX/XX
TIME: XX:XX
```

The two manual dump files released with the 2.3.7 system are defined as follows:

Unit	Name	ID	Length	Description
3	DUMP	D1	FFFF	TAPE DUMP FILE 1
3	DUMP	D2	FFFF	TAPE DUMP FILE 2

SYSTEM INSTALLATION PARAMETERS 5

System installation parameters are set in the operating system source code by running a full update against the appropriate program libraries.

This chapter includes descriptions of the primary system installation parameters, which, because of what they specify, are most often changed during installation of VSOS, and the secondary system installation parameters, that are seldom changed.

PRIMARY INSTALLATION PARAMETERS

MACHINE SIZE

This section describes the modifications that must be made if the site has other than the default memory size or number of vector pipes.

The installation parameters must be defined as shown in table 5-1.

Table 5-1. Machine Size Parameters (Primary)

Name	Description	Range/Limits	Default Value	Ident Seqnum	Deck Name	Deck Location
IP_BLOCKS	Number of words required in central memory for page table	2048, 4096, or 8192 words	2048	CH509.1	IPARAMS	SOSPL
IP_4PCY205	Number of vector pipes: 0 2 Pipes 1 4 Pipes	0 or 1	0	RA645A.63	IPARAMS	SOSPL
IP_MEM_SZE	Memory size in millions of words	1, 2, 4, 8, 16, or 32	1	CH509.2	IPARAMS	SOSPL

It is also necessary to specify the origin in the load statement for the resident system, as follows (RESCD.13 in BLDPL):

Memory Size	2 Pipe	4 Pipe
1 million	#78000	N/A
2 million	#98000	#E0000
4 million	#100000	#100000
8 million	#100000	#100000
16 million	#100000	#100000
32 million	#140000	#1C0000

DISK CONFIGURATION

The size of the permanent file index file (Q5IFI) must be increased if more than 12 disks are configured in the system. The size of the file must be #40 for each disk configured. The size of Q5IFI is specified in the AUTOCON CONFIG file.

The size of the virtual system paging file (Q5VSF) must be increased if more than 16 disks are configured in the system. The following formula must be used to determine the correct size of this nonsegmented, nonextendable file:

```
len = #500 + (#40 * number of disks)
```

The system checkpoint file is now a predefined file (refer to this manual and the VSOS Version 2 Operator's Guide). This file must be contiguous. The file size depends on the length of the virtual system paging file.

```
length (checkpoint file) = length (paging file) + #108
```

For the released system, the length of Q5VSF is #900 blocks and the length of the checkpoint file is #A08 blocks. If the length of the virtual system paging file is increased from the released value, then the length of the system checkpoint file must be increased by defining a new contiguous file under the checkpoint user number IP_CHECK_USER (released value - 000098).

WORKLOAD-RELATED INSTALLATION CONSIDERATIONS

The use of a system shared library (SHRLIB) at a site is workload-related. The following recommendations pertain to installing the system for various workloads. The following workload characterizations refer to the greatest percentage of work, or the highest priority work being run on the system. Some experimentation using these recommendations as starting points is probably necessary to determine the optimal shared library environment.

- If your site is primarily running large-machine sized jobs, SHRLIB should probably not be active. If SHRLIB is active because the large controllees are using the system library in SHRLIB, the minimum and maximum working set for SHRLIB should be set to 64 and 128 respectively.
- If your site is primarily running FTN200 compile and execute jobs, SHRLIB should be active and FTN200 included as a shared utility. Wherever possible when running jobs, use the GO option of FTN200. It is not necessary to lock down SHRLIB, but set the working set maximum to 384 blocks. If the maximum working set specified is too small, the system sends a message to the operator K display indicating that the working set must be increased. The system then automatically adjusts the working set of SHRLIB up or down to a value consistent with the use of SHRLIB.
- If the site is primarily executing controllees made up of FTN200 subroutines but does not execute the FTN200 compiler often, SHRLIB should be active but the FTN200 compiler need not be included as a shared utility. This decreases the size of SHRLIB.
- If an increase in the maximum working set for SHRLIB is not a problem, load shared utilities statically to get the best performance.
- Use of SHRLIB shows the most improvement on a one-million-word system with the SLGEN parameter TSP set to 1. On a two-million-word system the TSP should be set to 4. The SHRLIB improvement on this system is cut in half. On a four-million-word system, the TSP should be set to 16. The SHRLIB improvement is cut in half again. Generally, there is less SHRLIB improvement on larger machines because there is less paging activity.

JOB CATEGORIES

When the system is initially released, two job categories, JDEFAULT and INTRACTV, are defined as follows:

Parameter	JDEFAULT	INTRACTV
Maximum working set (blocks)	MAXWS	MAXWS
Maximum time limit (seconds)	99999	99999
Maximum large page limit	MAXWS/128	MAXWS/128
Maximum priority	2	14
Default priority	2	14
Maximum job count	15	50
Maximum executing batch job per user	5	-
Category Class	DFLT	DFLT

Adding Job Categories

You can add job categories by inserting code at line JCINIT.85 (deck JCINIT in SOSPL). The first line of each set of code must define the table entry ordinal from 1 to n. Refer to table 5-2 for more information on creating additional job categories. The general principle of high priority, low resources; low priority, high resources should be followed.

Table 5-2. Job Category Parameters (Primary) (Sheet 1 of 2)

Name	Description	Range/Limits	Default Value
CLASS†	Job category class.	1 to 4 ASCII characters	None
DP	Default priority.	1 to 15 (must be $\leq$ MP)	0
DFTL	Default time limit in seconds.	0 to 599940	0
JCAT	Job category name.	1 to 8 ASCII characters	None
MJC	Maximum job count.	1 to 50	0

† CLASS names allow all categories belonging to a class to be affected by INPT, ON/OFF operator commands. CLASS names should not be the same as the first four characters of a JCAT.

Table 5-2. Job Category Parameters (Primary) (Sheet 2 of 2)

Name	Description	Range/Limits	Default Value
MJU	Maximum execution batch job count.	0 to MJC	0
MP	Maximum priority.	1 to 15	0
MXLP	Maximum large page limit in blocks.	0 to (MAXWS/128)	0
MXTL	Maximum time limit in seconds.	0 to 599940	0
MXWS	Maximum working set in blocks.	0 to MAXWS	0
IP_JCAT_WS††	Interactive workstation job category.	1 to 8 ASCII characters	0

†† This parameter is applicable only when it is used with the separate, extra cost Remote Workstation Interface (RWI) product set.

Reordering Job Categories

Follow these steps to reorder or redefine the job categories:

1. Delete all job categories, except INTRACTV and JDEFAULT, from each user's list of valid job categories. Refer to the VSOS 2 Operator's Guide for a description of the procedure to follow for modifying the user directory using the EDITUD utility.

2. Reorder or redefine the job categories.

3. Run EDITUD again to add the appropriate job category permissions for each user.

Priority Map Table

The Priority Map Table (PMT) is a shared table that translates input queue priorities into execution class and level. This table is 15 words long and is indexed from 1 to 15. The index into the table is the user-supplied (or job category supplied in the case of user default) input queue priority. Interactive tasks default to priority 14 in the system as released. Each word of the table contains two, 32-bit quantities. The upper 32-bit field (known as C_PMT_CLASS) contains the class in which a task or job with the associated priority executes. The lower 32-bit field (known as C_PMT_LEVEL) contains the level within the class at which the job or task executes.

The valid range of values for C_PMT_LEVEL is from 1 to 15. If the class or level of any PMT entry is not within the valid range of values, the associated input queue priority is disabled. Any job that requests a disabled priority is aborted with an INVALID PRIORITY message.

Refer to table 5-3 for the release values of the PMT. Installations may change the contents of the PMT by modifying data statement set values in local tables, CLASS_VALUES and LEVEL_VALUES in deck JCINIT, lines JCINIT.37 through JCINIT.39, and JCINIT.42 through JCINIT.43, respectively.

Table 5-3. Priority Map Table

Entry (Priority)	Class	Level	Entry (Priority)	Class	Level
1	S_JOB	1	9	B_JOB	8
2	B_JOB	1	10	B_JOB	9
3	B_JOB	2	11	B_JOB	10
4	B_JOB	3	12	B_JOB	11
5	B_JOB	4	13	B_JOB	12
6	B_JOB	5	14	I_JOB	14
7	B_JOB	6	15	H_JOB	15
8	B_JOB	7			

The valid range of values for C_PMT_CLASS is from 1 to 4 (there is an additional class value of 5, which is reserved for system tasks). These values are known by class mnemonic, defined as follows.

Value	Class	Meaning
1	S_JOB	Standby
2	B_JOB	Batch
3	I_JOB	Interactive
4	H_JOB	High priority

FILES/STORAGE

Table 5-4 shows the installation parameter that relates to files and storage.

Table 5-4. Files/Storage Parameter (Primary)

Name	Description	Range/Limits	Default Value	Ident Seqnum	Deck Name	Deck Location
FILEREP	Retention period of a file, in days.	0 to 32767	30	SYSPAR.68	SYSPAR	SOSPL

MAGNETIC TAPES

Table 5-5 shows the installation parameters relating to magnetic tape.

Table 5-5. Magnetic Tape Parameters (Primary)

Name	Description	Range/Limits	Default Value	Ident Seqnum	Deck Name	Deck Location
IP_TPCM	Default conversion mode: 1 ASCII 2 EBCDIC	1 or 2	1	IPARAMS.66	IPARAMS	SOSPL
IP_TPDENS	Default tape density: 1 6250 GE 2 1600 PE	1 or 2	1	IPARAMS.65	IPARAMS	SOSPL
IP_TPACS	Batch access to tapes: 0 Batch access to tapes not allowed. 1 Batch access to tapes allowed.	0 or 1	1	IPARAMS.12	IPARAMS	SPSPL
IP_TPINT	Interactive access to tapes: 0 Interactive access to tapes not allowed. 1 Interactive access to tapes allowed.	0 or 1	1	IPARAMS.11	IPARAMS	SPSPL
IP_TPNSL	Nonstandard tape processing: 0 Processing inhibited; only ANSI standard labels are processed. 1 Users may read or write nonstandard labels.	0 or 1	0	IPARAMS.70	IPARAMS	SOSPL
IP_TPTF	Default tape format: 1 Large block (LB). 2 SCOPE internal (SI). 3 NOS internal (I). 4 Variable (V). #C NV format (NV).	1 to 4 or #C	1	CE406.1	IPARAMS	SOSPL
IP_TPULA	Unlabeled access to labeled tape: 0 Access disallowed for nonprivileged jobs. 1 System requests operator to determine if a nonprivileged job is allowed access.	0 or 1	0	CD422.1	IPARAMS	SOSPL

PERFORMANCE

Table 5-6 shows installation parameters that can be set to change system performance.

Table 5-6. Performance Parameters (Primary)

Name	Description	Range/Limits	Default Value	Ident Seqnum	Deck Name	Deck Location
IJOB_INIWS	Initial working set size for interactive tasks, in blocks.	16 to 256	32	MISSCH.162	MISSCH	SOSPL
IP_LALT	Length of alternator table.	32 or 63	32	CH496.1	IPARAMS	SOSPL
IP_MXBJU	Limits the number of batch jobs in execution for a user.	1 to 512	10	C3024.2	IPARAMS	SOSPL
IP_MXNJU	Limits the number of jobs in the input queue for a user.	1 to 512	256	C3641.2	IPARAMS	SOSPL
IP_SCED_ENABLE	Enables automatic start up of SCHEDULE for dynamic job scheduling.	0 to 1	1	C3005A.6	IPARAMS	SOSPL
IP_SHRB	The minimum number of faults allowed per 30 seconds for SHRLIB before the working set will be decreased.	0 to IP_SHRB	1*30	CH015B.2	IPARAMS	SOSPL
IP_SHRF	The maximum number of faults allowed per 30 seconds for SHRLIB before the working set will be increased.	IP_SHRB to #FFFF	8*30	CH015B.1	IPARAMS	SOSPL
IP_SHRI	The number of blocks to increment/decrement the SHRLIB working set when it is adjusted.	16 to 128	32	CH015B.3	IPARAMS	SOSPL

MISCELLANEOUS

Table 5-7 shows the installation parameters that do not fall into any of the other categories.

Table 5-7. Miscellaneous Parameters (Primary)

Name	Description	Range/Limits	Default Value	Ident Seqnum	Deck Name	Deck Location
EXCLAM	Break character.	N/A	1R!	C3178A.83	TTYMES	SOSPL
IP_DFRSTRT	If set, prevents restart of dropfiles that have time limit aborted.	0 or 1	0	CH819.1	IPARAMS	SOSPL
IP_DB	ITFS trace file generation (used for debugging only): 1 No trace. -1,0 Trace file generated.	-1, 0, or 1	1	IF_CPIP.19	IF_CPIP	SOSPL
IP_RMT_OPRMG	If set, indicates that remote operator is to receive begin/end job messages.	0 or 1	1	IPARAMS.55	IPARAMS	SOSPL
IP_TTYSC	Special character (SC).	1R! to 1R&	1R$	CH321.1	IPARAMS	SOSPL

REMOTE HOST FACILITY (RHF)

Table 5-8 shows the installation parameters that affect Remote Host Facility (RHF) features.

Table 5-8. RHF Parameters (Primary)

Name	Description	Range/Limits	Default Value	Ident Seqnum	Deck Name	Deck Location
IP_ATRAC	Trace is for the most recent file transfer only (default) or for the entire transfer.	-1, 1	-1	RA786.3	PARAMS	RHFPL
IP_MTBS	Requested block size for data transfers in groups of eight. If set to zero, data transfers use LCN default block sizes for binary transfers; 3840 for character transfers.	0 to 4064	4064	RA819.4	PARAMS	RHFPL
IP_REW_TRC	Used exclusively for DUMPF or LOADPF. If set to 1, trace file is rewound whenever a file is transferred. Otherwise, trace file will not be rewound.	0, 1	1	CF098.1	PARAMS	RHFPL
IP_TRACE	Trace file is turned on if IP_TRACE=1. Trace file is turned off if IP_TRACE=0.	0, 1	0	CH321.1	PARAMS	RHFPL

AUTOCON

Table 5-8.1 shows the installation parameters that affect AUTOCON features.

Table 5.8.1 AUTOCON Parameters (Primary)

Name	Description	Range/Limits	Default Value	Ident Seqnum	Deck Name	Deck Location
IP_SITE_ID	Site identifier.	1 to 8 ASCII characters	8HSN501_G	C3603.6	IPARAMS	SOSPL
IP_SYSSEL	Prompt for operator system selection.	0, 1	0	C3603.7	IPARAMS	SOSPL
IP_AFILE	Name of file of AUTOCON's config file names.	1 to 8 ASCII characters	8HQ5AUTOS	C3603.8	IPARAMS	SOSPL
IP_AUSER	Owner of IP_AFILE.	000000 to 999999	6R000000	C3603.9	IPARAMS	SOSPL
IP_LPLIMIT	Maximum large page limit.	See IP_MEM_SZE.	IP_MEM_SZE *16	C3603.10	IPARAMS	SOSPL
IP_MAXMEM	Maximum central memory size.	See IP_MEM_SZE.	IP_MEM_SZE	C3603.11	IPARAMS	SOSPL
IP_VRFN	Variable rate accounting file name.	See IP_VRFNAME.	IP_VRFNAME	C3603.12	IPARAMS	SOSPL
IP_VRUN	Variable rate accounting file owner.	See IP_VRFOWNER.	IP_VRFOWNER	C3603.13	IPARAMS	SOSPL

REMOTE WORKSTATON FACILITY (RWF)†

PIDs, LIDs, and NADs associated with each workstation network are installed via RWF installation parameters, instead of through the more usual AUTOCON parameters. Up to four RWF LIDs can be defined, each representing one CYBER 205 connection to a single RWI server serving a network of RWI client workstations.

An RWF interactive job category can also be defined that will allow separate control of RWF interactive sessions. If set to zero, RWF interactive sessions will be controlled by the standard INTRACTV job category.

Table 5.8.2 Remote Workstation Facility Parameters

Name	Description	Range/Limits	Default Value	Ident Seqnum	Deck Name	Deck Location
IP_WSN	Number of RWI servers directly connected to the CYBER 205 that serve a network of RWI client workstations. If non-zero, associated PIDs, LIDs, and NADs must be installed using the following installation parameters. (Integer)	0 to 4	0	CH910A.14	RHFTABS	SOSPL
IP_WSP1	Three-character PID associated with RWI servers 1 through 4. Installed in conjunction with RWF LIDs and NADs. (ASCII, left-justified, blank-filled)		NUL	CH910A.16	RHFTABS	SOSPL
IP_WSP2				CH910A.17	RHFTABS	SOSPL
IP_WSP3				CH910A.18	RHFTABS	SOSPL
IP_WSP4				CH910A.19	RHFTABS	SOSPL
IP_WSL1	Three-character LID associated with RWI servers 1 through 4. Installed in conjunction with RWF PIDs and NADs. (ASCII, left-justified, blank-filled)		NUL	CH910A.21	RHFTABS	SOSPL
IP_WSL2				CH910A.22	RHFTABS	SOSPL
IP_WSL3				CH910A.23	RHFTABS	SOSPL
IP_WSL4				CH910A.24	RHFTABS	SOSPL
IP_WSN1	NAD associated with RWI servers 1 through 4. Installed in conjunction with RWF PIDs and LIDs. Note that NAD numbers installed here must match the RWD NAD numbers installed at SYST time on the MCU. (Integer)	01 to #FF	0	CH910A.26	RHFTABS	SOSPL
IP_WSN2				CH910A.27	RHFTABS	SOSPL
IP_WSN3				CH910A.28	RHFTABS	SOSPL
IP_WSN4				CH910A.29	RHFTABS	SOSPL

† RWF (Remote Wordstation Facility) is part of RWI (Remote Workstation Interface), the software for workstation connections. The necessary software and hardware to support a workstation connection are products of ETA Systems, Inc., and are sold and distributed separately.

USER IDENTIFICATION

Table 5-9 shows the installation parameters you can use to change user identification.

Table 5-9. User Identification Parameters (Primary)

Name	Description	Range/Limits	Default Value	Ident Seqnum	Deck Name	Deck Location
IP_ADM_OPRID	Site security administrator user number in binary.	IP_MIN_OPRID to IP_MAX_OPRID	999996	CH717A.1	IPARAMS	SOSPL
IP_ADM_AOPRID	Site security administrator user number in ASCII.	IP_MIN_OPRID to IP_MAX_OPRID	999996	CH717A.2	IPARAMS	SOSPL
IP_MAXUSR	Maximum number of users allowed.	1 to 65535	4096	RA645A.89	IPARAMS	SOSPL
IP_PASS_REQ	Validity of password: 0 Blank (default) password is valid. 1 Blank password is not valid.	0 or 1	0	PASSCODE.78	PASSCODE	SOSPL
IP_PRIMOP	Primary operator user number in binary.	1 to 999999	000098	IPARAMS.49	IPARAMS	SOSPL
IP_A_PRIMOP	Primary operator user number in ASCII.	6R000001 to 6R999999	6R000098	CF246.1	IPARAMS	SOSPL
IP_SYS_PASS	System user password.	1 to 8 ASCII characters	SYSTEM	IPARAMS.25	IPARAMS	SOSPL
SYSAC	Account number for User-1.	1 to 8 ASCII characters	404SOV	SYSPAR.41	SYSPAR	SOSPL

SECONDARY INSTALLATION PARAMETERS

This section describes the secondary parameters. The released default settings are recommended; however, the site may choose to modify them.

ACCOUNTING

Table 5-10 shows installation parameters that can be used for accounting purposes.

Table 5-10. Accounting Parameters (Secondary) (Sheet 1 of 5)

Name	Description	Range/Limits	Default Value	Ident Seqnum	Deck Name	Deck Location
CPFACT	Percent charge for CPU.	0 to 100	100	ASINIT.97	ASINIT	SOSPL
DISCACM	Microseconds used for one disk access (read/write).	1 to 2^{20}-DISCSECM	38000	TIMEST.11	TIMEST	SOSPL
DISCSECM	Microseconds used to transfer one 512-word block to/from disk.	1 to 2^{20}-DISCACM	1000	TIMEST.12	TIMEST	SOSPL
EXFACT	Percent charge for explicit I/O.	0 to 100	0	ASINIT.99	ASINIT	SOSPL

Table 5-10. Accounting Parameters (Secondary) (Sheet 2 of 5)

Name	Description	Range/Limits	Default Value	Ident Seqnum	Deck Name	Deck Location
IMFACT	Percent charge for implicit I/O.	0 to 100	0	ASINIT.48	ASINIT	SOSPL
IP_F_VR	0 Do regular accounting. 1 Do variable rate accounting.	0 or 1	0	C3153A.1	IPARAMS	SOSPL
	NOTE If IP_F_VR is set to 1, the LOAD statement for the virtual system must contain the parameter GRSP = *T VRFX.					
IP_VRFNAME	Variable rate file name.	1 to 8 ASCII characters	Q5VRF	C3602.1	IPARAMS	SOSPL
IP_VRFOWNER	Variable rate file name owner.	0 to 999999	999998	C3602.2	IPARAMS	SOSPL
IP_TWGT_AVWSS	Average working set size during this accounting period.	0 to $2^{28}-1$	0	TCHARGE.44	TCHARGE	SOSPL
IP_TWGT_LLPC	Lost large page count.	0 to $2^{28}-1$	-156000	TCHARGE.46	TCHARGE	SOSPL
IP_TWGT_LSPC	Lost small page count.	0 to $2^{28}-1$	-29000	TCHARGE.45	TCHARGE	SOSPL
IP_TWGT_LPACCI	Disk accesses (I/O requests) for large page implicit writes during this accounting period.	0 to $2^{28}-1$	28000	TCHARGE.36	TCHARGE	SOSPL
IP_TWGT_LPACCX	Disk accesses (I/O requests) for large page explicit reads and writes during this accounting period.	0 to $2^{28}-1$	28000	TCHARGE.35	TCHARGE	SOSPL
IP_TWGT_LPGFLT	Number of large page faults (disk accesses and I/O requests) during this accounting period.	0 to $2^{28}-1$	156000	TCHARGE.41	TCHARGE	SOSPL
IP_TWGT_MEMU	Memory used during this accounting period; working set size * CPU time.	0 to $2^{28}-1$	0	TCHARGE.31	TCHARGE	SOSPL

Table 5-10. Accounting Parameters (Secondary) (Sheet 3 of 5)

Name	Description	Range/Limits	Default Value	Ident Seqnum	Deck Name	Deck Location
IP_TWGT_SCPU	System CPU time in microseconds used during this accounting period.	0 to $2^{28}-1$	1	TCHARGE.30	TCHARGE	SOSPL
IP_TWGT_SPACCI	Disk accesses (I/O requests) for small page implicit writes during this accounting period.	0 to $2^{28}-1$	28000	TCHARGE.38	TCHARGE	SOSPL
IP_TWGT_SPACCX	Disk accesses (I/O requests) for small page explicit reads and writes during this accounting period.	0 to $2^{28}-1$	28000	TCHARGE.37	TCHARGE	SOSPL
IP_TWGT_SPGFLT	Number of small page faults (accesses for faults) during this accounting period.	0 to $2^{28}-1$	29000	TCHARGE.42	TCHARGE	SOSPL
IP_TWGT_SPSECI	Number of disk sectors transferred for implicit writes during this accounting period.	0 to $2^{28}-1$	1000	TCHARGE.40	TCHARGE	SOSPL
IP_TWGT_SPSECX	Number of disk sectors transferred for explicit reads and writes during this accounting period.	0 to $2^{28}-1$	1000	TCHARGE.39	TCHARGE	SOSPL
IP_TWGT_TAPACC	Number of tape accesses (I/O requests issued) for reads and writes during this accounting period.	0 to $2^{28}-1$	4000	C1644.5	TCHARGE	SOSPL
IP_TWGT_TAPFNT	Number of nonread and nonwrite tape functions during this accounting period.	0 to $2^{28}-1$	250	C1644.6	TCHARGE	SOSPL
IP_TWGT_TAPWDS	Number of 16-bit bytes transferred to or from tape files during this accounting period.	0 to $2^{28}-1$	3	C1644.4	TCHARGE	SOSPL
IP_TWGT_UCPU	User execution CPU time in microseconds used during this accounting period.	0 to $2^{28}-1$	1	TCHARGE.29	TCHARGE	SOSPL
IP_TWGT_VSCALLS	Number of virtual system user calls made during this accounting period.	0 to $2^{28}-1$	0	TCHARGE.43	TCHARGE	SOSPL

Table 5-10. Accounting Parameters (Secondary) (Sheet 4 of 5)

Name	Description	Range/Limits	Default Value	Ident Seqnum	Deck Name	Deck Location
IP_TWGT_WS2SM	Working set too small.	0 to $2^{28}-1$	0	TCHARGE.47	TCHARGE	SOSPL
IP_VRICALL	Limit of system message #000E option 7 calls that can be issued per task.	200	200	IPARAMS.40	IPARAMS	SOSPL
IP_WGT_AVWSS	Weight for average working set.	0 to $2^{28}-1$	0	MCHARGE.52	MCHARGE	SOSPL
IP_WGT_LLPC	Weight for lost large page count.	0 to $2^{28}-1$	0	MCHARGE.53	MCHARGE	SOSPL
IP_WGT_LSPC	Weight for lost small page count.	0 to $2^{28}-1$	0	MCHARGE.54	MCHARGE	SOSPL
IP_WGT_LPACCX	Weight for disk accesses for large page explicit reads and writes.	0 to $2^{28}-1$	0	MCHARGE.43	MCHARGE	SOSPL
IP_WGT_LPACCI	Weight for disk accesses for large page implicit writes.	0 to $2^{28}-1$	0	MCHARGE.44	MCHARGE	SOSPL
IP_WGT_LPGFLT	Weight for large page faults.	0 to $2^{28}-1$	0	MCHARGE.49	MCHARGE	SOSPL
IP_WGT_MEMU	Weight for memory use.	0 to $2^{28}-1$	0	MCHARGE.38	MCHARGE	SOSPL
IP_WGT_SCPU	Weight for system CPU time used by user.	0 to $2^{28}-1$	0	MCHARGE.42	MCHARGE	SOSPL
IP_WGT_SPACCI	Weight for disk accesses for small page implicit writes.	0 to $2^{28}-1$	0	MCHARGE.46	MCHARGE	SOSPL
IP_WGT_SPACCX	Weight for disk accesses for small page explicit reads and writes.	0 to $2^{28}-1$	0	MCHARGE.45	MCHARGE	SOSPL
IP_WGT_SPGFLT	Weight for small page faults.	0 to $2^{28}-1$	0	MCHARGE.50	MCHARGE	SOSPL
IP_WGT_SPSECI	Weight for disk sectors transferred for implicit writes.	0 to $2^{28}-1$	0	MCHARGE.48	MCHARGE	SOSPL
IP_WGT_SPSECX	Weight for disk sectors transferred for explicit reads and writes.	0 to $2^{28}-1$	0	MCHARGE.47	MCHARGE	SOSPL

Table 5-10. Accounting Parameters (Secondary) (Sheet 5 of 5)

Name	Description	Range/Limits	Default Value	Ident Seqnum	Deck Name	Deck Location
IP_WGT_TAPACC	Weight for tape accesses (I/O requests issued) for reads and writes.	0 to $2^{28}-1$	0	MCHARGE.40	MCHARGE	SOSPL
IP_WGT_TAPFNT	Microseconds used for one tape function (read/write).	0 to $2^{28}-1$	0	MCHARGE.41	MCHARGE	SOSPL
IP_WGT_TAPWDS	Microseconds to transfer one, 16-bit byte to/from tape.	0 to $2^{28}-1$	0	MCHARGE.40	MCHARGE	SOSPL
IP_WGT_UCPU	Weight for user CPU.	0 to $2^{28}-1$	1	MCHARGE.37	MCHARGE	SOSPL
IP_WGT_VSCALLS	Weight for number of virtual system calls.	0 to $2^{28}-1$	0	MCHARGE.51	MCHARGE	SOSPL
IP_WGT_WS2SM	Weight for working set too small.	0 to $2^{28}-1$	0	MCHARGE.55	MCHARGE	SOSPL
MEFACT	Percent charge for memory.	0 to 100	0	ASINIT.98	ASINIT	SOSPL
REFACT	Percent charge for remote I/O.	0 to 100	0	ASINIT.102	ASINIT	SOSPL
SYFACT	Percent charge for system.	0 to 100	0	ASINIT.101	ASINIT	SOSPL
TPACCM	Microseconds used for one tape access (read/write).	1 to $2^{28}-1$	4000	C1644.2	TIMEST	SOSPL
TPFUNCTM	Microseconds used for one tape function (nonread/ write).	1 to $2^{28}-1$	250	C1644.3	TIMEST	SOSPL
TPWDSM	Microseconds to transfer one 16-bit byte to/from tape.	1 to $2^{10}-1$	3	C1644.1	TIMEST	SOSPL

FILES/STORAGE

Table 5-11 shows installation parameters that relate to files and storage.

Table 5-11. Files/Storage Parameters (Secondary) (Sheet 1 of 2)

Name	Description	Range/Limits	Default Value	Ident Seqnum	Deck Name	Deck Location
EXTSIZ	Maximum percentage of creation file length to extend file.	0 to (2-1*100)	50	SYSPAR.43	SYSPAR	SOSPL
IP_AF_LENG	Size of accounting file, in 512-word blocks.	1 to LDSK	100	RA645A.54	IPARAMS	SOSPL
IP_AF_NAME	Name of primary (active) accounting file.	1 to 8 ASCII characters	Q5AF	RA645A.52	IPARAMS	SOSPL
IP_AF_USER	Owner of accounting file.	1 to 8 ASCII characters	999998	C2920.1	IPARAMS	SOSPL
IP_AF_BUSER	Binary user number.		999998	C2920.2	IPARAMS	SOSPL
IP_AF2_NAME	Name of secondary accounting file.	1 to 8 ASCII characters	Q5AF2	RA645A.53	IPARAMS	SOSPL
IP_DISKALARM	Disk usage alarm.	1 to 99	85	C3856.1	IPARAMS	SOSPL
IP_DUMFILE	Maximum default file size, in 512-word blocks, for private files.	1 to LDSK	65535	RA645A.103	IPARAMS	SOSPL
IP_IFI_NAME	Name of inactive file index (IFI) file.	1 to 8 ASCII characters	Q5IFI	RA645A.49	IPARAMS	SOSPL
IP_LFILEI	Number of FILEI entries allowed.	3072 to 6144	4096	CG498A.27	IPARAMS	SOSPL
IP_MXPFILE	Maximum file size, in 512-word blocks, for pool files.	1 to LDSK	65535	RA645A.102	IPARAMS	SOSPL
IP_N_DFSIZE	Minimum drop file size (unless user specifies drop file size using CDF load option).	2 to 65535	37	RA645A.98	IPARAMS	SOSPL
IP_PFILENMAX	Maximum length of the PFI file on a disk in 512-word blocks.	32 or 64	64	RA645A.94	IPARAMS	SOSPL

Table 5-11. Files/Storage Parameters (Secondary) (Sheet 2 of 2)

Name	Description	Range/Limits	Default Value	Ident Seqnum	Deck Name	Deck Location
IP_POOLS_LENG	Length of pool directory in blocks.	1 to 9	18	CH958A.2	IPARAMS	SOSPL
IP_POOLS_NAME	Name of pool directory.	1 to 8 ASCII characters	Q5POOLST	CH958A.1	IPARAMS	SOSPL
IP_SDF_NAME	Name of primary (active) dayfile.	1 to 8 ASCII characters	Q5SDF	RA645A.55	IPARAMS	SOSPL
IP_SDFSIZE	Size of dayfile in 512-word blocks.	1 to LDSK	256	RA645A.57	IPARAMS	SOSPL
IP_SDF2_NAME	Name of secondary dayfile.	1 to 8 ASCII characters	Q5SDF2	RA645A.56	IPARAMS	SOSPL
IP_SDF_USER	Owner of system dayfile.	1 to 8 ASCII characters	999998	C2920.3	IPARAMS	SOSPL
IP_SDF_BUSER	Binary user number.		999998	C2920.4	IPARAMS	SOSPL
IP_UD_NAME	Name of user directory.	1 to 8 ASCII characters	Q5UDF	C2089A.1	IPARAMS	SOSPL
LDFS	Maximum drop file length in sectors.	1 to LDSK	#1FFFF	RA645A.331	SYSPAR	SOSPL
LDSK	Maximum single file length in sectors.	1 to size of largest device set	976563	RA645A.329	SYSPAR	SOSPL

LOAD

The LOAD utility limits the number of entry points, externals, and common blocks. The limit may be changed by changing the parameter defined in table 5-12 and by rebuilding the LOAD utility.

Table 5-12. LOAD Parameters (Secondary)

Name	Description	Released Value	Ident Seqnum	Deck Name	Deck Location
COMXRL	Number of common blocks * 2.	18000 for 9000 entries	CH146.3	LDXREF	SPSPL
ENTXRL	Number of entry points * 3 + 3.	9003 for 3000 entries	CH146.1	LDXREF	SPSPL
EXTXRL	Number of externals * 2.	18000 for 9000 entries	CH146.2	LDXREF	SPSPL

MAGNETIC TAPES

Table 5-13 shows the installation parameters relating to magnetic tapes.

Table 5-13. Magnetic Tape Parameters (Secondary) (Sheet 1 of 2)

Name	Description	Range/Limits	Default Value	Ident Seqnum	Deck Name	Deck Location
IP_TPVA	Default tape volume accessibility character. A blank (#20) means unlimited access to information on the tape.	0 to #FF	#20	IPARAMS.75	IPARAMS	SOSPL
IP_TPVOL	Volume 1 label writing: 0 Only privileged users are allowed to write VOL1 labels. 1 Users are allowed to write VOL1 labels.	0 or 1	0	IPARAMS.73	IPARAMS	SOSPL
IP_TPVSN	Operator override of a job's VSN and assignment of tapes without matching VSNs: 0 Operator cannot override. 1 Operator can use j.VSN, vsn command to override job's VSN; operator can use j.ASSIGN command to assign matching VSNs.	0 or 1	1	IPARAMS.71	IPARAMS	SOSPL

Table 5-13. Magnetic Tape Parameters (Secondary) (Sheet 2 of 2)

Name	Description	Range/Limits	Default Value	Ident Seqnum	Deck Name	Deck Location
IP_TPFA	Default tape file accessibility character; indicates any restrictions on who may access the information in the file. This value corresponds to the FA parameter of the LABEL control statement. A blank (#20) means unlimited access; any other character means all future accesses to the tape must specify this character as the FA parameter. File accessibility is not checked for privileged jobs; if FA is A (#41), the first six characters of the owner id field must match the user number. The remaining eight characters of the field must be blanks.	0 to #FF	#20	IPARAMS.76	IPARAMS	SOSPL
IP_TPRU	Read past end-of-information on tapes: 0 Users not allowed to read past EOI. 1 Users allowed to read past EOI.	0 or 1	0	IPARAMS.64	IPARAMS	SOSPL
IP_TPEC	Hardware error correction mode for GCR tapes: 0 Hardware error correction enabled; system allows single-track errors to be written if they can be corrected when the tape is read. 1 Standard error recovery processing.	0 or 1	0	IPARAMS.72	IPARAMS	SOSPL
IP_TPEXP	Writing to tapes with unexpired labels: 0 Users cannot write on tape with unexpired label. 1 Operator can override unexpired label and allow users to write on tape.	0 or 1	0	IPARAMS.80	IPARAMS	SOSPL

PERFORMANCE

Table 5-14 shows installation parameters that can change the system performance.

Table 5-14. Performance Parameters (Secondary)

Name	Description	Range/Limits	Default Value	Ident Seqnum	Deck Name	Deck Location
OBTBLNO	Maximum number of modules on an object library file.	1 to 6000	6000	RC085.3	UTOLE	SPSPL
MAXMODS	Maximum number of modules per library.	1 to 6000	6000	RC085.2	UTOLE	SPSPL
AJOB_INIWS	Initial working set size for system tasks in blocks.	16 to 256	32	MISSCH.163	MISSCH	SOSPL
MAX_HOSTS	Maximum number of connected hosts of ITFS.	1 to 32	32	IF_CPIP.47	IF_CPIP	SOSPL
MAX_TERMS	Maximum number of ITFS interactive terminals.	1 to 128	128	IF_CPIP.46	IF_CPIP	SOSPL
NRESBLKS	Number of blocks (in multiples of 16) reserved for the nonpageable portion of the resident system.	N/A	6*16	RA645A.450	PAGE	SOSPL
IP_RJMJI	Flag to indicate if the remote host should resubmit batch job if user has reached the maximum number of jobs in the input queue (IP_MXNJU). If IP_RJMJI=1, the remote host is signalled to resubmit the batch job. If IP_RJMJI=0, the batch job is evicted with an error notice sent to the user. For some remote hosts (e.g., NOS), if signalled to resubmit the batch job, the remote host sends the same job immediately, thus blocking jobs from entering the CYBER 200 until one of the user's jobs moves from the input queue.	0,1	1	C3641.4	IPARAMS	SOSPL

REMOTE HOST FACILITY (RHF)

RHF system shared tables T_APPT, T_CAT, T_CRT, T_RHMFT, and T_RHFT are defined by the parameters shown in table 5-15.

Table 5-15. RHF Table Size Parameters (Secondary)

Name	Description	Default Value	Ident Seqnum	Deck Name	Deck Location
IP_N_ACTIVE	The maximum number of RHF applications (combination of all types except ITFS)† allowed to be concurrently active for the system. The maximum value allowed is 32.	32	RHFTABS.6	RHFTABS	SOSPL
IP_N_APPL	The highest number of servicer type applications (QTFS, PTFS, DLFS or SCFS)† on any remote host that are called from the local host.	3	RHFTABS.4	RHFTABS	SOSPL
IP_N_LIDS	The total number of logical identifiers (LIDs) defined for the RHF network that are addressable from the system.	64	RHFTABS.3	RHFTABS	SOSPL
IP_N_LOCALAPP	The maximum number of types of local RHF applications (not including ITFS). There are five types currently possible: QTF, QTFS, PTF (MFLINK), PTFS, or DLF†.	6	RHFTABS.5	RHFTABS	SOSPL
IP_N_PIDS	The total number of host processors (PIDs) addressable on the RHF network.	14	CG486.1	RHFTABS	SOSPL

† ITFS = Interactive Transfer Facility Servicer
QTFS = Queue File Transfer Facility Servicer
PTFS = Permanent File Transfer Facility
DLFS = Dump/Load Facility Servicer
SCFS = Status and Control Facility Servicer
QTF = Queue File Transfer Facility
PTF = Permanent File Transfer Facility
DLF = Dump/Load Facility

SYSTEM INTERFACE LANGUAGE (SIL)

Table 5-16 shows the installation parameters that affect SIL routines.

Table 5-16. SIL Parameters (Secondary)

Name	Description	Range/Limits	Default Value	Ident Seqnum	Deck Name	Deck Location
IP_BT	Blocking type.	2 characters	2	IPARAM.16	IPARAM	LIBPL
IP_MNR	Minimum record length.	0 to #FFFF	0	IPARAM.19	IPARAM	LIBPL
IP_MXR	Maximum record length.	0 to #FFFF (zero indicates no maximum record length)	0	IPARAM.20	IPARAM	LIBPL
IP_PC	Padding character.	Any ASCII character	" " (#20)	IPARAM.23	IPARAM	LIBPL
IP_RMK	Record mark delimiting character (for R-type records).	Any ASCII character	#1F	IPARAM.25	IPARAM	LIBPL
IP_RT	Record type.	0 W-type (control word) 1 F-type (ANSI fixed length) 2 R-type (record mark) 7 U-type (undefined)	2	IPARAM.32	IPARAM	LIBPL
IP_SC_SIL	Operating system special character.	Any ASCII character	"$" (#24)	IPARAM.33	IPARAM	LIBPL
IP_SFO	File organization.	0 (sequential)	0	IPARAM.18	IPARAM	LIBPL
IP_SRL_CALLIMIT	The number of times routine Q5CPUTIM can be called from a program.	0 to $2^{47}-1$	10	Q5CPUTIM.65	Q5CPUTIM	LIBPL

SYSTEM DELIMITERS

Table 5-17 shows the system delimiters that can be defined for each installation.

Table 5-17. System Delimiters Parameters (Secondary)

Name	Description	Range/Limits	Default Value	Ident Seqnum	Deck Name	Deck Location
IP_N_DELIMS	Number of installation-defined delimiters.	1 to 9	9	IPARAMS.30	IPARAMS	SOSPL
IP_1DELIM	Installation system-defined delimiter.	0 to #FF	"("	IPARAMS.31	IPARAMS	SOSPL
IP_2DELIM	Installation system-defined delimiter.	0 to #FF	","	IPARAMS.32	IPARAMS	SOSPL
IP_3DELIM	Installation system-defined delimiter.	0 to #FF	"/"	IPARAMS.33	IPARAMS	SOSPL
IP_4DELIM	Installation system-defined delimiter.	0 to #FF	"="	IPARAMS.34	IPARAMS	SOSPL
IP_5DELIM	Installation system-defined delimiter.	0 to #FF	"+"	IPARAMS.35	IPARAMS	SOSPL
IP_6DELIM	Installation system-defined delimiter.	0 to #FF	"-"	IPARAMS.36	IPARAMS	SOSPL
IP_7DELIM	Installation system-defined delimiter.	0 to #FF	")"	IPARAMS.37	IPARAMS	SOSPL
IP_8DELIM	Installation system-defined delimiter.	0 to #FF	"."	IPARAMS.38	IPARAMS	SOSPL
IP_9DELIM	Installation system-defined delimiter.	0 to #FF	" "	IPARAMS.39	IPARAMS	SOSPL

USER IDENTIFICATION

Table 5-18 shows the installation parameters that can be used to change user identifications.

Table 5-18. User Identification Parameters (Secondary)

Name	Description	Range/Limits	Default Value	Ident Seqnum	Deck Name	Deck Location
IP_CEUSERNO	CE diagnostic user number.	100 to 999999	999980	CG618.1	IPARAMS	SOSPL
IP_N_DEFSEC	Default security level.	1 to 8	1	IPARAMS.26	IPARAMS	SOSPL
IP_MAX_OPRID	Maximum system user number (binary).	100 to 999999	999998	IPARAMS.52	IPARAMS	SOSPL
IP_MIN_OPRID	Minimum system user number (binary).	100 to 999999	999990	IPARAMS.51	IPARAMS	SOSPL
IP_SCED_UN	User number for system job SCHEDULE.	1 to 99	30	C3005A.5	IPARAMS	SOSPL

UTILITIES

Table 5-19 shows the installation parameters that can be used to change utilities.

Table 5-19. Utilities Parameters (Secondary) (Sheet 1 of 2)

Name	Description	Range/Limits	Default Value	Ident Seqnum	Deck Name	Deck Location
IP_DAYF	Dayfile first option: 0 Option is turned off. Dayfiles will remain last in the output file (normal processing). 1 Option is turned on. Dayfiles will be moved to the front of the output file or will be the first record of the output file.	0 or 1	0	C3095.3	PARAMS	RHFPL
IP_SCF	Determines whether PTFS will process status and control commands. If 3, PTFS may execute control commands (DROP, and DIVERT) or the status command (Q). If 2, PTFS may execute control commands. If 1, PTFS may execute status commands. If 0, PTFS will not execute any status and control commands.	0 to 3	3	CH850.1	PARAMS	RHFPL

Table 5-19. Utilities Parameters (Secondary) (Sheet 2 of 2)

Name	Description	Range/Limits	Default Value	Ident Seqnum	Deck Name	Deck Location
IP_U_LEN	Default file size.	1 to #FFFF	8	UTIPAR.5	UTIPAR	SPSPL
IP_U_RECALL	The number of minutes to wait before the ATTACH utility tries to gain access to a file that is in use with nonshareable access.	1 to 30	1	UTIPAR.7	UTIPAR	SPSPL
IP_U_NA	The total number of minutes the ATTACH utility continues retrying before returning an error code.	1 to 60 (a multiple of IP_U_RECALL)	10	UTIPAR.8	UTIPAR	SPSPL
IP_LPP	Default lines per page.	0 to #FF	0	C3774A.1	IPARAMS	SOSPL
IPQSTPW	Determine whether QSTATUS will place password information file. If 1, then QSTATUS will place the key word "PW" into the keyword record and put password information (in hashed form) into the status records. If 0, then the keyword "PW" is not put into the keyword record and password information is not placed into the status records.	0 or 1	1	QSCFITP.14	QSCFIPT	RHFPL

TOOLS 6

This chapter describes additional tools and utilities supplied with the system that are useful during the build process or for maintaining the system.

ACC3

Use ACC3 to analyze the system accounting file.

Format:

```
ACC3,input,TAPE5=input,output,TAPE6=output.
```

input — Specifies the file where the directives exist. If omitted, ACC3 will use the directives on the file INPUT.

output — Specifies the file to which output from ACC3 is to be written. If omitted, ACC3 will write to the file OUTPUT.

Input directives:

All directives are input as keyword=value or keyword=value1/value2/.../valuen.

Keyword	Value	Meaning
TYPE	IQM	Accounting file is IQM. Default is IQM.
POOL	aaaaaaaa	Name of the pool that contains accounting files. Default is " ".
INC	nnn	Usage time width internal. Default is 10.
JOBSTAT	Y/N	If Y, job statistics are printed. Default is N.
SPECIAL	Y/N	If Y, special task statistics are printed. Default is N.
USAGE	Y/N	If Y, usage statistics are printed. Default is N.
SYSTEM	Y/N	If Y, system statistics and a system resource snap are printed. Default is N.
JOBSNAP	Y/N	If Y, job snap records are printed. Default is N.
IOSTAT	Y/N	If Y, I/O statistics are printed. Default is N.
RHFSTAT	Y/N	If Y, RHF statistics are printed. Default is N.
FILES	aaaaaaaa/a/.../a	Up to 10 files or search keys may be input.
END		Terminates directive processing.

Summary information is put into the output file. Billing information is put on file TAPE1.

ASDF

Use ASDF to generate an abbreviated version of the system dayfile, listing RHF errors for the current day. ASDF must be executed under user number 999998, where current system dayfiles are present (permanent or local). To get the latest information for the current day, the operator must perform a TMSF,DF prior to using ASDF.

Format:

```
ASDF.
```

When the analyzer has completed execution, the following output files will be local for the user to route to a remote printer:

Q5DRHF	Dayfile summary of all RHF errors.
Q5DQTF	Dayfile summary of QTF errors.
Q5DQTFS	Dayfile summary of QTFS errors.
Q5DPTFS	Dayfile summary of PTFS errors.
Q5DPTF	Dayfile summary of MFLINK/PTF errors.

BADSPOT

Use BADSPOT to display and optionally modify the bad sector map of a disk unit.

Format:

```
BADSPOT,dn
```

dn Device number whose bad sector map is to be displayed/modified.

1. BADSPOT displays the current bad sector map entries and then prompts you with:

   ```
   ADD/DELETE SECTORS (A,D,-S TO STOP)
   ```

2. If the user enters A or D, BADSPOT responds with:

   ```
   ENTER SECTOR ADDRESS(HEX)
   ```

   The user then enters the sector address to be added or deleted from the bad sector map.

3. BADSPOT then asks:

   ```
   MORE?
   ```

   If the user enters Y, then BADSPOT requests the next sector address to be added or deleted. If the user enters N, BADSPOT displays the first prompt message again. If the user has completed all additions and deletions, he/she enters S.

4. If modifications to the bad sector were made, BADSPOT displays the new bad sector map and asks:

   ```
   REWRITE BAD SECTOR MAP?
   ```

   If the user wishes the bad sector map to be updated on the disk, then he/she enters Y. Note that the changes to the bad sector map do not take effect until the next AUTOLOAD of VSOS. At that time it is possible for a Permanent File Verification error 26 (overlap between a file and the bad sector map) to occur.

BEGIN

Use BEGIN to interactively execute control statement procedure files in the same manner as the BEGIN command to the batch processor.

Format:

```
BEGIN,pname,pfile,p1,p2,...,pn.
```

Refer to the detailed explanation of the BEGIN statement and parameters in VSOS Version 2 Reference Manual, Volume 1.

BLDIFILE

BLDIFILE may be used to build the input file for QSTATUS (see QSTATUS description in this section). The calling format for BLDIFILE is as follows:

```
BLDIFILE,lfn,lid,"string1","string2",..."stringn".
```

lfn	The name of the QSTATUS input file.
lid	The RHF logical identifier of the remote host.
$string_1$ - $string_n$	The directives for MFLINK to send to the remote host to save the QSTATUS information file. A maximum of 10 strings can be specified.

Each call to BLDIFILE appends directives to the specified lfn. The following is a sample job submitted from a NOS front-end to build the QSTATUS input file. The example assumes the CYBER 205 has a LID of CY2, and that there are 3 front-ends with LIDs of NOS, NBE, and IBM.

```
BIFILEJB,ST=CY2.
USER,U=qstusernum,PA=password.
BLDIFILE,qstifile,NOS,"USER,user number,password,family",
   "REPLACE,MFSTCY2","PERMIT,MFSTCY2/AC=R".
BLDIFILE,qstifile,NBE,"ACCOUNT,dept number,account number.",
   "PURGE,MFSTCY2,ID=id,LC=1.","CATALOG,MFSTCY2,ID=id.".
BLDIFILE,qstifile,IBM,"REPLACE,DSN=MFSTCY2,UNIT=unit,LRECL=80.".
PURGE,qstifile.
DEFINE,qstifile.
```

qstusernum	The privileged user number from which the QSTATUS job will execute.
qstifile	The name of the QSTATUS input file.

Upon completion of the above BIFILEJB job, the contents of the qstifile file should be as follows:

```
ST=NOS.
JCS="USER,user number,password,family","REPLACE,MFSTCY2",
"PERMIT,MFSTCY2/AC=R".
ST=NBE.
JCS="ACCOUNT,dept number,account number.",
"PURGE,MFSTCY2,ID=id,LC=1.","CATALOG,MFSTCY2,ID=id.".
ST=IBM.
JCS="REPLACE,DSN=MSTCY2,UNIT=unit,LRECL=180".
```

CHECK

CHECK is a utility used to determine file ownership attributes. For each specified file, a message is sent to the job dayfile or interactive terminal.

Format:

```
CHECK( filelist )

    filelist      list of file names (required; no default)
```

Messages:

filename CONNECTED

The file is connected to an interactive terminal.

filename LOCAL

The file is a local file.

filename NOT FOUND

The file is not attached.

filename PERMANENT

The file is an attached permanent file.

filename poolname (POOL)

The file belongs to the pool poolname.

filename PUBLIC

The file belongs to the public file set.

filename TAPE

The file is assinged to a tape drive.

filename - UNKNOWN DEVICE TYPE: xxxxxxxx

An error occurred during processing of the file.

NO FILES SPECIFIED

At least one file name must be specified.

unexpected SIL response

An error occurred during processing of the file.

For virtual code files, the load date, time, and version are appended to the applicable messages.

CVPOOL

CVPOOL is a utility which will convert the pool directory file from the 256 entry format (pre-VSOS 2.3.7) to the 512 entry format (VSOS 2.3.7).

Format:

CVPOOL,inpdf,outpdf

inpdf — Lfn of the input pool directory file. The file may be a local, unattached private, pool, or public file and it must be of the 256 entry type. If not specified, the default is Q5POOLS.

outpdf — Lfn of the output pool directory file. If a local file exists with the same name, it will be returned. This file will be of the 512 entry format. If not specified, the default is Q5POOLST.

CVPOOL generates a listing of all pools found in the input pool directory file, the poolboss for each pool, and a list of the users that have access to the pool. This listing file is in the file OUTPUT.

DELIVER

Use DELIVER to generate an enlarged header page on an OUTPUT listing. Each DELIVER call generates the designated header messages on two successive sheets.

Format:

DELIVER,header1,header2,header3.

Each of the header parameters is optional. They can be ASCII letters or digits (1 through 8). The - character is printed as a blank. Each header will be printed as a separate enlarged line.

DISKS

Provides information on configured disks.

Usage:

DISKS

Example output:

	DEV	STAT	ALLOC	DEV SET	USAGE	ERRS	TYPE	DAV
PACK01	42	ON	SYS	DVSTOA	86%	0	8190	4
PACK02	47	OFF	SYS	DVSTOB	97%	0	8190	4
PACK03	41	ON	SYS	DVSTOA	85%	0	8190	4
PACK04	52	ON	PRI/DROP	DVSTOC	14%	0	8190	64

EDIT

EDIT is a line-oriented editor with string search capabilities. A HELP command is available if you need a command summary.

Format:

```
EDIT,I=input file name (required), O=output file name (default=EDITOUT),SEQ=start point
      in Edit file (default=100),BY=increment(default=10)
```

Command syntax (L1 and L2 are line numbers):

Syntax	Description
L1=text	Add or replace L1
L1,L2=text	Auto add or replace L1, L1+L2, etc. ("=" ends add mode)
L1-	Delete L1
L1,L2-	Delete L1 through L2
L1	List L1
L1,L2	List L1 through L2
Q	Quit execution; exit without copying
C	Copy edit text to output file
L	Display last line of file
E	Copy file and terminate execution
/STRING/	Locate first line containing string
/STRING/,L1	Locate first line after L1 with string
N	Locate next occurrence of string
TX=N1,N2	Set tab character X at N1,N2
T	Clears tab
R	Resequence text
NONREDUCE	Don't reduce the output file
H	Displays command syntax

FILEI

FILEI is a utility used to obtain hex display of the contents of a file index entry.

All specified files are listed in the FILE INDEX TABLE format described in the CYBER 200 OS Reference Manual, Volume 2.

Format:

```
FILEI,lfn-list,PUBLIC=lfn-list,PRIVATE=lfn-list,POOL=poolname,lfn-list,L=lfn
```

lfn-list	List of private files with FILEI information listed.	
PUBLIC=lfn-list	Public files listed:	
	lfn-list	List of 1 to 255 public file names separated by commas.
POOL=poolname,lfn-list	Pool files listed:	
	poolname	Name of pool to which user is attached.
	lfn-list	List of 1 to 255 file names in a pool poolname. If omitted, all files in a pool are listed.
PRIVATE=lfn-list	Private files listed:	
	lfn-list	List of 1 to 255 private file names, separated by commas. * indicates all private files are listed.
L=lfn	Name of file to which output is written. Output file name must be 1 through 8 letters or digits, beginning with a letter. Default in batch mode is OUTPUT. Default in interactive mode returns output to terminal. FILEI control statement format.	

Omission of all parameters is equivalent to FILEI(PRIVATE=*).

FIND

FIND is a utility for the investigation of UPDATE old program libraries. It is used primarily to locate specified text strings in an oldpl file.

Usage:

```
FIND(P=oldpl,L=lfile,I=ifile)
```

oldpl — UPDATE old program library file to be scanned.

Default: OLDPL

lfile — Output file name.

Default: OUTPUT

ifile — Command input file name.

Default:

Batch: INPUT

Interactive: terminal

The following commands can be used in the command input file.

Command	Description
*ALL	Selects all decks for processing and initiates the search sequence described below. Command entry resumes with an empty deck list.
*COUNT	Lists the total number of active cards in the decks specified (by deck list) or the total number of active cards in the oldpl file (if the deck list is empty). Command entry resumes with an empty deck list.
*DECKS	Lists the oldpl file *DECK/*COMDECK names (except for YANK$$$). Command entry resumes with an empty deck list.
*END	Completes a command sequence. If the deck list is empty, FIND terminates. If the deck list is not empty, the search sequence described below is executed. Command entry resumes with an empty deck list.

Command	Description
*LIST	Prints active cards. The card text is listed to the right of its identifier and sequence number. The behavior of this command is determined by the number of deck names listed. Command entry resumes with an empty deck list. If no decks are specified (the deck list is empty), the deck name and all active cards are listed for each deck. If a single deck is specified, one of the following ranges must be specified. Terminal input You are prompted with: ENTER START SEQ NO. (IDENT.SEQ) OR ALL - Enter *ALL to list the entire deck. - Enter xxxxxxxx.nnnn to identify the first card in the list range; then you are prompted with: ENTER END SEQ NO - Enter xxxxxxxx.nnnn the last card in the list range. File input Use an *ALL card to list the entire deck. Use one xxxxxxxx.nnnn card to identify the first card in the list range and one xxxxxxxx.nnnn card to identify the last card in the list range. If the initial xxxxxxxx.nnnn card does not specify an active card in the selected deck, no cards will be listed. If the final xxxxxxxx.nnnn card does not specify an active card in the selected deck, the listing ends with the last active card in the deck.
*SEARCH	Initiates the search sequence described next, under Search Sequence. If the deck list is empty, all the decks are processed. If the deck list is not empty, processing is restricted to the specified decks. Command entry resumes with an empty deck list.
deckname	Any command entry that is not one of the command words: *ALL, *COUNT, *DECKS, *END, *LIST, or *SEARCH is assumed to be a deck name. Deck names are added to the deck list and affect the processing of commands. There is an internal restriction of 20 deck names in the deck list. If more than 20 deck names are specified, the following message is displayed: DECK LIST OVERFLOW - DECK NAME IGNORED The deck name is not added to the deck list.

SEARCH SEQUENCE

The *ALL, *END, and *SEARCH commands invoke search processing.

The following sequence is executed if the input is from a file.

1. The search string is read. The search string is 80 characters of text before trailing blanks are trimmed.

   If the search string is null or entirely blank, FIND terminates.

2. The specified decks, or all decks if the deck list is empty, are scanned for active cards with text that matches the search string. For each scanned deck in which a match occurs, the deck name and all active cards containing the search string are listed.

   Trailing blanks can be specified by appending a @ to the search string.

3. Command entry resumes with an empty deck list.

The following verification steps are executed before the oldpl file is scanned if the input is from a terminal.

1. If the deck list is not empty, the text DECKS ENTERED, and the deck list names are first displayed.

2. You are then prompted with:

   ENTER SEARCH STRING

   Enter the search string. The search string is 80 characters of text before trailing blanks are trimmed.

   If the search string is null or entirely blank, you are prompted again.

3. When a valid search string is read, you are prompted with:

   VERIFY SEARCH STRING

   The trimmed search string is displayed.

4. You are prompted with:

   ACCEPT? (Y OR N)

   Enter Y to continue processing as in batch mode.

   Enter N to return to the ENTER SEARCH STRING prompt.

   Any other response returns to the ACCEPT? (Y OR N) prompt.

IN

IN is an input record generator. It creates records on the file INPUT.

Each parameter of the IN command will generate one record in the output file.

Any parameters within single quotes are placed directly into the output file. All special characters may be included within this literal value. Blanks may precede other characters in the literal to position the data in the correct column. This is a simple way of generating text files for any purpose.

Any parameters not within single quotes are preceded by *C in the output text file. This is a simple way of generating input files for use with the UPDATE command. Periods and underbars may be used within these parameters.

Example of an IN statement:

```
IN,DECK1,DECK2,DRANGE.F_RANGE,'A TYPICAL
LITERAL','      PARAMETER',DECK3)
```

The IN statement produces the file INPUT with the following information:

```
*C DECK1
*C DECK2
*C DRANGE.F_RANGE
A TYPICAL LITERAL
      PARAMETER
*C DECK3
```

IOQ7LIB

The library IOQ7LIB consists of FTN200 subroutines that use a double buffering scheme to read and write disk files in a sequential manner. The library uses the Q7BUFIN, Q7BUFOUT, and Q7WAIT concurrent input/output (I/O) routines.

The I/O library uses the concurrent I/O routines Q7BUFIN (input) for the user calls SINIT and SGET, Q7BUFOUT (output) for the user calls SPUT and SPUTF, and Q7WAIT for I/O completion and returned status. The library can manage up to eight files.

The library uses a double buffer scheme with as many as 21 logical records in a buffer. The number of records in a buffer is dependent upon the number of words in a record. The buffer size and the number of records per physical I/O are selected from two tables, LRSATBL and IORTBL. Table 6-1 shows the relationship between the two tables.

The tables are in common block USRSETUP and can be changed by the user. Refer to User Selection of Physical I/O in this chapter for more information.

Table 6-1. Relationship between LRSATBL and IORTBL Tables

Table LRSATBL (Record Size)	Table IORTBL (Records Per I/O Buffer)
1 through 511	21
512 through 1365	14
1366 through 2730	7
2731 through 5461	6
5462 through 10922	3
10923 through 16384	2
16385 through 65535	1

All files managed by the library must be requested or defined prior to use with record type U.

No FORTRAN I/O calls can be used on the files managed by the library.

The library files are written by SPUT and read by SINIT and SGET.

The buffer size is determined from the logical record size supplied by the user during the library initialization call (SINIT). From 1 to 21 logical records can be written as one physical record. There are no gaps or other devices to differentiate one physical record from the next. For example, if the size of a logical record is 6279 (64-bit) words, then each physical record contains 3 logical records (currently IORTBL is 3 for this record size) or 18837 words of data and 107 words of nonrecord data. The nonrecord data occurs because the disk is read or written in 512-word blocks. (For additional information, refer to the concurrent I/O subroutines in the FORTRAN 200 Reference Manual.)

Each time a logical record switches from a read SGET to a write SPUT state, it must first call SINIT or SREWIND. When switching from a write to a read state, SPUTF should be called before SINIT or SREWIND is called.

If I/O errors occur, an error message is sent to the dayfile and the output file. The job is then terminated.

To access these routines, you must:

- Attach the pool containing the IOQ7LIB library using PATTACH.
- Then use the library parameter (LIBRARY=) and the grouping parameter (GROL=) on the LOAD card, as follows:

```
LOAD(routine,LIBRARY=IOQ7LIB,GROL=*IOBFS).
```

If user common blocks are grouped, the I/O common block IOBFS should be grouped last; neither USRSETUP nor BFSDATA need be grouped. The I/O common block is large enough for eight files (16 large pages); however, only the number of pages required for active files are mapped into the drop file.

SUPPORTIVE SUBROUTINES

The following subroutines are part of IOQ7LIB.

SINIT

SINIT is called to initialize the necessary variables to access the specified units. SINIT must be called at least once. Additional calls to SINIT are required if one of these conditions occurs:

- A new file (logical unit number) is to be accessed.
- The record size of a file changes.
- Changes to common block USRSETUP were made after a call to SINIT.

You are responsible for ensuring that all output buffers have been emptied by a call to SPUTF.

The calling sequence is:

```
CALL SINIT(lrl,ntu,ntr,iua,info).
```

lrl	An array of record lengths, in words, one array element per logical file, specified in array iua.
ntu	The total number of logical files that are used during this sequence.
ntr	The number of logical files that are to be read by this call; zero if none.
iua	An array of logical file numbers that are used during this sequence, one array element per logical file.
info	A nonzero print indicator that lists information about the logical files. The printout occurs prior to returning from this call. The printout consists of the following for each logical unit: record size, pages I/O, record I/O, and both virtual bit addresses.

SINIT can be used to fill the double buffers of each logical file that is to be initially read during this sequence. This allows the first SGET for each logical file to return the data immediately and not wait for the I/O to complete. If SINIT is to read in from any logical files, the parameter ntr must be set to the number of logical files to be read, and the array iua must have the first ntr entries as the read files. The record lengths in array lrl must correspond to the files described in the array iua.

SGET

SGET is called each time a record of data is required by the user.

The calling sequence is:

```
CALL SGET(iux,array)
```

iux	The file number to get data from disk.
array	The user's buffer address into which the data is put.

SPUT

SPUT is called each time a record of data is to be written to a file. Several records may be buffered before the actual I/O occurs on the file.

The calling sequence is:

```
CALL SPUT(iux,array)
```

iux	The file number to put data to disk.
array	The user's buffer address from which data is taken and put into the file.

SPUTF

SPUTF is called to write out the remaining records that have been buffered up waiting to be written to the file. This call is made to ensure the data integrity of the unit. This call should be made before any additional calls to SINIT or any calls to SREWIND are made.

The calling sequence is:

```
CALL SPUTF { 0   }
           { iux }
```

0 Indicates that all files are to be logically rewound and their respective buffers marked as empty.

iux File number whose current I/O buffer should be written if it contains any data. The buffer for file iux is marked as empty.

SREWIND

SREWIND logically rewinds the files and marks the I/O buffers as empty. SREWIND can be used in place of any subsequent SINIT calls if all SINIT requirements are met. Refer to SINIT, earlier in this chapter.

The calling sequence is:

```
CALL SREWIND { 0   }
             { iux }
```

0 Indicates that all files are to be logically rewound and their respective buffers marked as empty.

iux File number of the file to be logically rewound. The buffer for file iux is marked as empty.

USER SELECTION OF PHYSICAL I/O

The library IOQ7LIB allows you to select the number of logical records per physical record. This selection allows for maximizing available buffer space.

The number of logical records in the physical I/O buffer is determined by the setting of two, eight-word entry tables in the common block USRSETUP. These tables are the LRSATBL and IORTBL tables.

The LRSATBL table contains the number of words per logical record in ascending order. The maximum number of words in a logical record and physical I/O buffer is 65535 words. You are free to change any or all eight entries; unused entries, if any, must be set to the value of the last usable entry.

Examples:

- Default values: 511, 1365, 2730, 5461, 10922, 16384, 65535, 65535. The last entry (eighth) is not used so it is set to the last used (seventh) entry.
- A user selection: 500, 1024, 2048, 4196, 10025, 10025, 10025, 10025. The last three entries are not used and set to the last used (fifth) entry.
- A user change: 511, 1365, 4096, 8192, 10922, 16384, 65535, 65535. Only the third and fourth entries were changed from the default values.

The IORTBL table contains the number of logical records in an I/O buffer (the number of record times the record size cannot exceed 65535 words).

Examples:

- Default value in LRSATBL:

  511, 1365, 2730, 5461, 10922, 16384, 65535, 65535.

- Default value of IORTBL:

  21, 14, 7, 6, 3, 2, 1, 1.

- Maximum value for IORTBL:

  128, 48, 24, 12, 6, 3, 1, 1.

The combination of the two tables determines the amount of data that is written (Q7BUFOUT) or read (Q7BUFIN) from each I/O buffer, rounded up to the nearest 512-word buffer.

Set up the common block USRSETUP containing the two physical I/O tables LRSATBL and IORTBL, as follows:

1. Initialize the tables with the FORTRAN DATA statement. When loading the compiler binary output and the IOQ7LIB subroutine, load last the routine that initializes the tables LRSATBL and IORTBL.

   Example:

   ```
   OLE(I=IOQ7LIB,USERBIN,M=NEWMM)
   LOAD(NEWMM,routine,GROL=*IOBFS)
   ```

   This method allows the user to set the tables in common block USRSETUP using a DATA statement.

2. The tables can be changed by the user at execution time prior to calling SINIT. SINIT must be called again for any further changes.

   Example:

   ```
   LOAD(USERBIN,LIBRARY=IOQ7LIB,routine,GROL=*IOBFS)
   ```

EXAMPLES

Figures 6-1, 6-2, and 6-3 show actual examples using IOQ7LIB.

```
00001         PROGRAM QEQNTL(TAPE1,TAPE2,TAPE3,,TAPE1=TAPE1,TAPE2=TAPE2,TAPE3=TAPE3)
00002         PARAMETER(LOOP1=3)
00003         PARAMETER(LEN=65536)
00004         COMMON/IOAR/A(LEN),B(LEN),C(LEN),D(LEN)
        C
        C COMMON BLOCK FOR PHYSICAL I/O,REFERENCE APPENDIX A
        C
00005         COMMON/USRSETUP/LRSATBL(8),IORTBL(8)
        C
        C DIMENSION ARRAYS FOR UNITS AND LENGTHS
        C
00006         DIMENSION IUA(5),LRX(5)
        C
        C INITIALIZE RECORD LENGTHS AND UNIT NUMBERS
        C
00007         DATA LRX(1),LRX(2),LRX(3)/3072,6279,10752/
00008         DATA IUA(1),IUA(2),IUA(3)/1,2,3/
        C
        C SET UP PHYSICAL I/O TABLES
        C  MUST BE SET UP AT EXECUTION TIME BECAUSE THE LIBRARY IS LOADED LAST
        C
00009         LRSATBL(4)=3072
00010         IORTBL(4)=5
00011         A=.11
00012         B=.22
        C
        C SET UP PARAMETERS FOR CALL TO SINIT:READ UNITS,TOTAL UNITS,PRINTOUT
        C
00013         NTR=0
00014         NTU=3
00015         INFO=1
00016         CALL SINIT(LRX,NTU,NTR,IUA,INFO)
00017         DO 500 I=1,LOOP1
        C
        C WRITE DATA TO THE FILES
        C
00018         CALL SPUT(IUA(1),A)
00019         CALL SPUT(IUA(2),B)
00020         C=A
00021         A=B
00022         B=C
00023    500  CONTINUE
00024         D=0.0
        C
        C WRITE LAST RECORD(IF ANY LEFT) TO THE UNITS
        C
00025         CALL SPUTF(0)
00026         DO 100 K=1,3
        C
        C REWIND THE UNITS SO THEY CAN BE READ
        C
```

Figure 6-1. Code Using IOQ7L1B (Sheet 1 of 2)

```
00027         CALL SREWIND(0)
00028         DO 1000 J=1,LOOP1
00029         A=0.0
00030         B=0.0
00031         C=0.0
        C
        C READ THE UNITS,FIRST CALL TO UNIT MUST WAIT TILL BOTH I/O BUFFERS FILL
        C
00032         CALL SGET(IUA(1),A)
00033         CALL SGET(IUA(2),B)
00034         C=A+B
        C
        C WRITE OUT THE SUM TO 1 UNIT
        C
00035         CALL SPUT(IUA(3),C)
00036    1000  CONTINUE
        C
        C WRITE LAST RECORD TO UNIT
        C
00037         CALL SPUTF(IUA(3))
        C
        C SET UP CALL TO SINIT TO READ THE 3 UNITS BUT NO PRINTOUT
        C
00038         NTR=3
00039         INFO=0
00040         CALL SINIT(LRX,NTU,NTR,IUA,INFO)
        C
        C GET DATA FROM EACH UNIT,NO WAIT FOR I/O - I/O INITIATED BY SINIT
        C
00041         CALL SGET(IUA(1),A)
00042         CALL SGET(IUA(2),B)
00043         CALL SGET(IUA(3),C)
00044         D=D+A+B+C
        C
        C REWIND ONE UNIT
        C
00045         CALL SREWIND(IUA(1))
        C
        C WRITE ON FIRST RECORD OF THE FILE
        C
00046         CALL SPUT(IUA(1),D)
        C
        C WRITE RECORD TO FILE
        C
00047         CALL SPUTF(IUA(1))
00048         PRINT 9,K,D(1),A(1),B(1),C(1)
00049    100  CONTINUE
00050         STOP
00051      9   FORMAT(/,' PASS ',I2,' D(1) IS ',F15.3,/
              1  ,' A(1) IS ',F5.2,' B(1) IS ',F5.2,' C(1) IS ',F5.2)
00052         END
```

Figure 6-1. Code Using IOQ7L1B (Sheet 2 of 2)

```
   R E C O R D   O F I/O   S P A C E   U S A G E          1


                PAGES/                    BIT ADDRESS OF
   UNIT    LRS     10      R/10       BUFFER 1          BUFFER  2

      1   3072      30       5    000001800000    0000018F0000

      2   6279      37       3    0000019E0000    000001C0000

      3  10752      63       3    000001D28000    000002000000
TOTAL LARGE PAGES USED      3
TOTAL SMALL PAGES USED  260
MAPPING IS            LARGE

PASS  1 D(1) IS            .660
A(1) IS   .11 B(1) IS   .22 C(1) IS   .33

PASS  2 D(1) IS           2.420
A(1) IS   .66 B(1) IS   .22 C(1) IS   .88

PASS  3 D(1) IS           7.700
A(1) IS  2.42 B(1) IS   .22 C(1) IS  2.64
```

Figure 6-2. IOQ7LIB Output

```
    RESOURCE(WS=1100,LP=8,TL=500)
    COMMENT.REQUEST OR DEFINE FILES WITH RECORD TYPE OF U
    REQUEST(TAPE1/150,RT=U)
    REQUEST(TAPE2/309,RT=U)
    REQUEST(TAPE3/525,RT=U)
    FTN200 (B=USERBIN,OPTIMIZE=1,LO=x)
    COMMENT.ATTACH THE POOL CONTAINING THE LIBRARY IOQ7LIB
    PATTACH(USERIO)
    COMMENT.SET UP PROPER LOAD CARD
    LOAD(USERBIN,LIBRARY=IOQ7LIB,CN=TESTIO/900,CDF=1200,GRLP=*IOAR,GROL=*IOBFS)
    TESTIO.
```

Figure 6-3. IOQ7LIB Control Statements

IOTST

Use IOTST to verify data paths between central memory and disks. IOTST uses the customer engineer (CE) tracks on the disks to write, read, and verify a pattern.

Format:

```
IOTST,ZIP=y,R=rc,PACKxx
```

y	The zip code number of a NAD that contains packs to which I/O is to be done.
rc	A repeat count to indicate the number of times each CE track is to be read and written.
PACKxx	The name of a pack to which the I/O is to be done.

ZIP=y or PACKxx must be specified if R=rc is specified.

The default is all disks with a repeat count of 1.

If the program is run interactively, the message:

```
ENTER PATTERN
```

is displayed. Enter up to 16 hexadecimal digits, which will be used as the pattern to write to disk. If you enter a period, the default pattern (#0000FFFFFFFFFFFF) is used.

If the program detects a pattern mismatch, it displays:

```
FILE COMPARE ERROR UNIT x
CETRzzyy PATTERN FOUND = pppppppppppppppp
```

x	Pack on which failure occurred.
CETRzzyy	CE file on which error occurred.
pppppppppppppppp	Data read from the file.

If the program is run interactively, the message:

```
RETRY, CONTINUE, OR STOP (R,C,S)?
```

is displayed. The R, C, and S in the message are defined as follows:

R	Write and read of the pattern to the failing file is reinitiated.
C	Program continues to the next CE file.
S	Program terminates.

LFNS

Use LFNS to list file information for up to 256 attached or pool files.

Format:

```
LFNS[,p1][,p2][,p3].
```

p1 — Parameter one lists all attached filenames with a given prefix. The default is to list all attached filenames.

p2 or p3 — Parameters two and three are used to search pool and system files and to list specific file information according to the following parameters.

POOL=poolname	Lists all files in POOL poolname.
SYSTEM=PUBLIC	Lists all public files.
SYSTEM=POOL	Lists all files in the system pool.
INFO=PNUM	Lists attached files and the pack number that they are on.
INFO=lfn	Lists length and pack number for file lfn.
INFO=LENGTH	Lists filenames and lengths of all attached files.
INFO=*	Lists filenames, lengths, and pack numbers for all attached files.

To list user filenames:

LFNS. Lists user files. The displayed filenames have their status indicated by the following prefixes:

;	Private, permanent file.
*	Attached, permanent file.
(blank)	Local file.

MERGE

Use MERGE to merge a number of ASCII files such as prestored files, list files, and so forth.

Format:

```
MERGE,lfn1,lfn2,...,lfnn,mergefile.
```

lfn1 Names of files.

lfnn To be merged.

Warning messages are issued for illegal or missing parameters.

NADUMP

Use NADUMP to analyze NAD dumps.

Format:

```
NADUMP(I=lfn,O=list)
```

lfn A local file containing the NAD dump.

list The name of the file to contain the output. The default is OUTPUT.

If a NAD dump is on a MCU pack, make it a local file by using READMCU.

Example:

```
NADUMP(I=DUMP6)
```

DUMP6 contains the dump for NAD 6. The output is on the file OUTPUT.

NAMEPACK DISK LABELING UTILITY

CAUTION

Use of this utility will destroy present contents of the diskpack.

This utility places a label on the diskpack and initializes the permanent file index (PFI) and directory of file segmentation (DFS). Entries for the following files are placed in the PFI.

CETR91nn - CE TRACK 1
CETR92nn - CE TRACK 2
CETR93nn - CE TRACK 3
LABELLnn - LABEL
PFInn - PFI
DFSnn - DFS
BADS81nn - BADSPOT

nn is the pack number.

Unused entries in the PFI have the pattern #000C1F1C000C1F1C. Unused entries in the DFS are zeroed.

LABELING

NAMEPACK labels a disk according to the information given by an interactive user. NAMEPACK checks for an existing label. Any existing label is assumed to be at sector address #230, #320, #410, #500, #5F0, #6E0, #7D0, #8C0, or #9B0. #230 is the system default. If a label exists, the system asks if it should be destroyed; if not, NAMEPACK terminates. Otherwise, the routine creates a label and a PFI entry for the label, the PFI, CE tracks, DFS, and a bad spot map. The label is created at #230. If an error occurs, an attempt is made to put the label at #320, #410, #500, #5F0, #6E0, #7D0, #8C0, or #9B0.

The label contains the volume name VOL3. The pack name is PACKxx, where xx is the pack number.

The PFI name is PFIxx, where xx is the pack number. The length of the PFI is set to the value entered by the user; single-density packs default to #1F and double-density packs to #40. Both the label and the PFI are created under the system user number and account. The bad sector map is created by NHCD diagnostics.

NOTE

When running NAMEPACK, the NADs must be logically up, but the disk that is being labeled must be logically down and off.

The following is an example of an interactive user session using NAMEPACK.

NAMEPACK Execute line that starts NAMEPACK running. Must be run under operator user number only.

```
ENTER PACK NUMBER IN HEXADECIMAL (#XX)
  LEGAL VALUES ARE #01 THRU #80.
```

#nn Any number between 1 and #80. The pack number is used to create the pack identifier PACKnn, and the files LABELnn, PFInn, DFSnn, BADS81nn, CETR91nn, CETR92nn, and CETR93nn, where nn equals the pack number.

```
ENTER DEVICE NUMBER IN HEXADECIMAL (#XX)
  LEGAL VALUE ARE #01 THRU #80.
```

#nn Any number between 1 and #80. The device number can be obtained from the E,D display (refer to figure 6-4).

DEV	LOG STAT	OWNER	DEVICE SET	PACK NAME	DEV TYPE	DAU	SPACE UTIL	ERROR
xxnn	nnnn	nnn	nnnnnn	nnnnnnnn	nnnn	nnn	nn%	nnn
xxnn	nnnn	nnn	nnnnnn	nnnnnnnn	nnnn	nnn	nn%	nnn

Column	Description
DEV	
xx	The equipment mnemonic (DA for disk).
nn	The device number, in two hexadecimal digits (from 01 to #80).
LOG STAT	Logical status (ON/OFF/DOWN).
OWNER	Pack ownership (SYS/PRI).
DEV TYPE	81922 (default).
DAU	Disk allocation unit.
SPACE UTIL	Space utilization. Specifies how much space is used on the device as a percent.

Figure 6-4. Example of E,D Display

```
TYPE OF DISKPACK:  81912 or 81922, or 887?
  81912 -- SINGLE DENSITY 18 SECTOR 819
  81922 -- DOUBLE DENSITY 18 SECTOR 819
  887   -- IHD DISK SUBSYSTEM
```

nnnn The type of diskpack to initialize.

```
ENTER DISK ALLOCATION UNIT VALUE IN DECIMAL.
  IF VALUE ENTERED IS NOT A MULTIPLE OF 4, IT
  WILL BE ROUNDED UP TO THE NEXT MULTIPLE OF 4.
   for 819:    MINIMUM = 4 ,  MAXIMUM = 180.
   for 887:    MINIMUM = 4 ,  MAXIMUM = 152.
```

nnn — The disk allocation unit (DAU) is a decimal number ranging from 4 to 180 for an 819.

```
ENTER LENGTH OF PFI IN HEXADECIMAL BLOCKS (#XX)
  DEFAULT IS: XXXXXXXX
```

#nn — The length of the PFI can be specified by entering nn. Entering a blank followed by a carriage return causes the default to be used. The default is #40 for 81922. The default is #20 for 81912.

```
ENTER LOCATION OF BAD SECTOR MAP IN HEXADECIMAL (#XXXXX)
  DEFAULT IS: XXXXXXXX
```

#nnnnn — The location of the bad sector map can be specified by entering nnnn. Entering a blank followed by a carriage return causes the default to be used. Default is #11F82 for an 819.

```
ENTER OPERATOR ACCOUNT NUMBER
  DEFAULT IS: XXXXXXXX
```

accnum — The account number under which the label, DFS, and PFI are created can be specified by entering accnum. Entering a blank followed by a carriage return causes 404SOV to be used.

```
ENTER CE ACCOUNT NUMBER
  DEFAULT IS: XXXXXXXX
```

accnum — The account number under which the bad sector map and CE tracks are created can be specified by entering accnum. Entering a blank followed by a carriage return causes 404SOV to be used.

```
ENTER DEVICE SET NUMBER IN HEXADECIMAL (#XX)
  LEGAL VALUES ARE #01 THRU #80.
```

nn — A hexadecimal value ranging from 1 to #80.

An any time during the input phase of NAMEPACK, the user may enter ABORT to terminate NAMEPACK.

```
DO YOU REALLY WANT TO RELABEL    XXXXXXXX?   (Y/N)
```

This prompt is displayed only if NAMEPACK has found an existing label on the pack.

xxx — To continue, the user must respond with YES; otherwise, NAMEPACK terminates.

ERROR MESSAGES

If the system displays any of the following error messages, your job has been terminated.

INVALID TYPE

Type is not 819.

PACKNAME APPEARS IN DISF TABLE

Pack must be logically off. Turn it off at AUTOLOAD time.

DRIVE, ZIP IN USE

Pack must be logically off. Turn it off at AUTOLOAD time.

CANNOT LOCATE BLK FOR LABEL AND READ IT.

Pack is bad in area of possible label.

ZIP NOT VALID

Zip code is not defined to system.

BAD FORMAT DRIVE, ZIP

Zip code or drive is zero.

BAD SEC MAP NOT READABLE

During sector location validation for initial pack files, the bad sector map could not be read from pack.

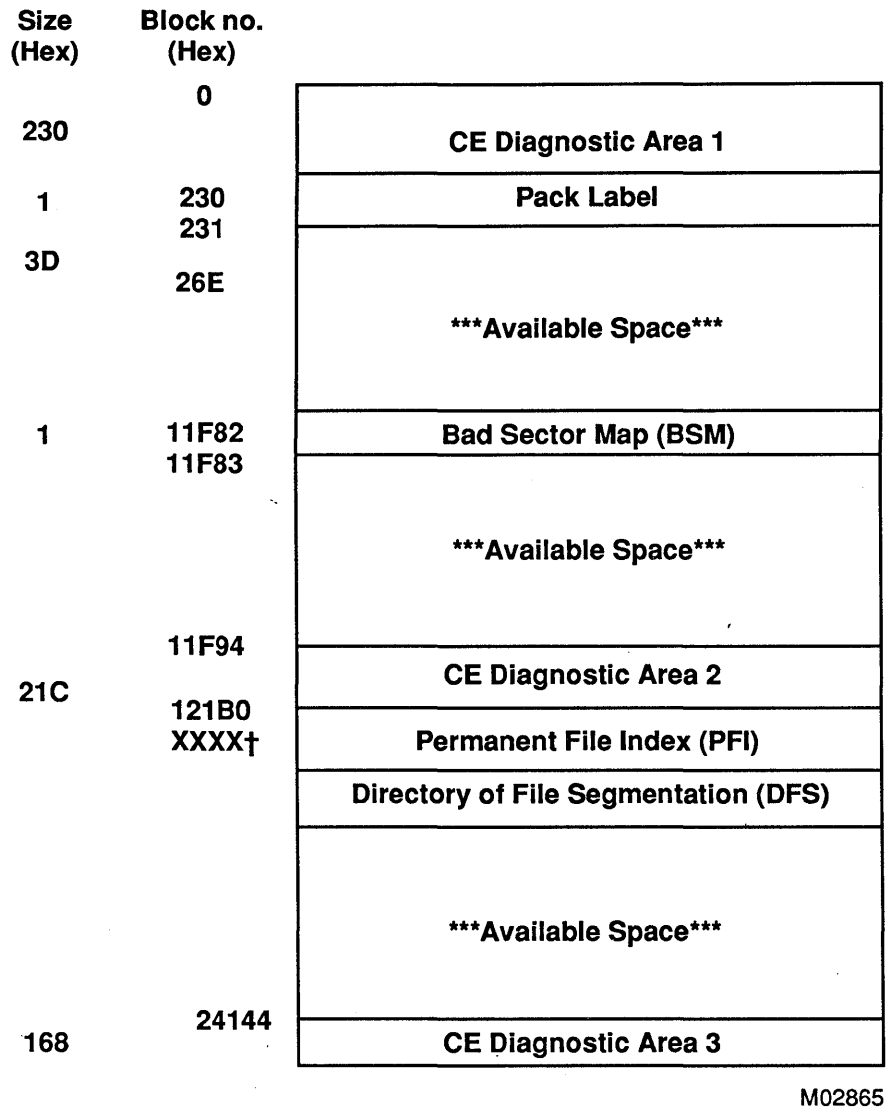
System interface language (SIL) routines can return error messages during NAMEPACK processing. Refer to the VSOS 2 Reference Manual, Volume 1, for an explanation of these messages.

The following error messages are output when an operating system problem, a hardware problem, or a problem in NAMEPACK code occurs. In these messages, xx is the response code returned by resident in response to a resident call, and nn is the debug trace number used in the specific resident call. Refer to the listing of NAMEPACK to determine the exact nature of the problem.

```
PROBLEM IN #F = #F004 MESSAGE
TCRC=xx TCFROM  nn

PROBLEM IN #F = #F003 MESSAGE
TCRC=xx TCFROM  nn

PROBLEM IN #F = #C500 MESSAGE
TCRC=xx TCFROM  nn

PROBLEM IN #F = #C501 MESSAGE
TCRC=xx TCFROM  nn
```

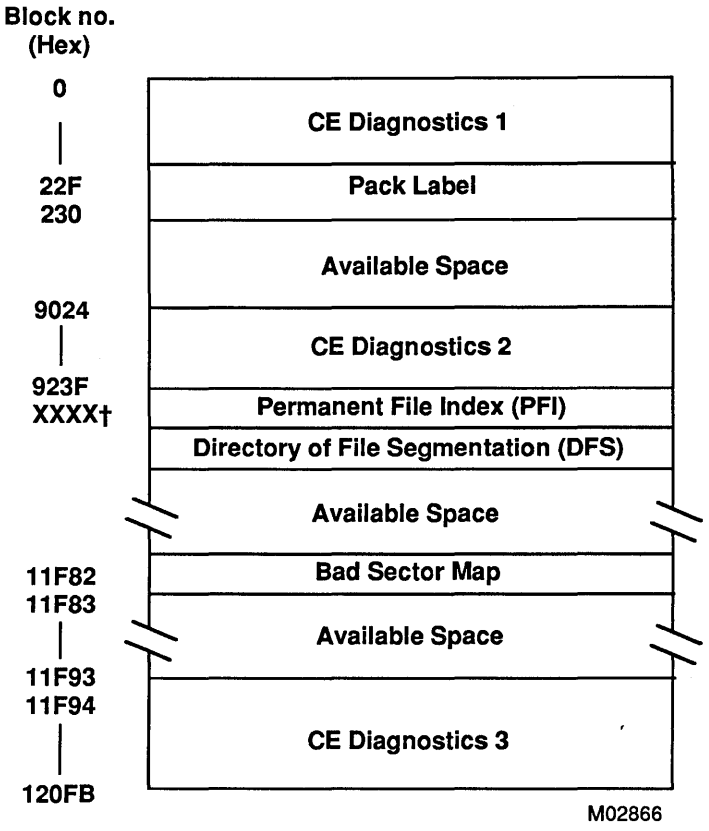
ORGANIZATION OF DISK PACKS

NAMEPACK organizes the disk packs as shown in figures 6-5 through 6-6.



†NAMEPACK places the PFI and DFS on a DAU boundary immediately following CE Diagnostic Area 2.

Figure 6-5. 81922 Disk Pack (Double-Density, 18-Sector) Format



†NAMEPACK places the PFI and DFS on a DAU boundary immediately following CE Diagnostic Area 2.

Figure 6-6. 81912 Disk Pack (Single-Density, 18-Sector) Format

PAGE

PAGE allows a user to examine a printable file page by page. PAGE is not an editor and does not alter the file being examined.

Format:

```
PAGE,lfn1,lfn2
```

lfn1 File to be examined. Default is OUTPUT.

lfn2 Print file. Default is PRINT.

When PAGE is ready to accept commands, it prompts:

```
READY..
```

As each command is processed, PAGE prompts for new input with ?.

PAGE displays information in two modes, single screen and full screen. In single-screen mode, only the first 64 characters of a line are displayed, and the page size is 16 lines. In full-screen mode, 135 characters of a line are displayed, and the page size is 8 lines.

To display the contents of a file, enter nnn. nnn is the line number within the file. PAGE begins displaying a page of information starting at line nnn.

These control commands are available to direct which page of a display is to appear:

Command	Description
*	Display the end-of-file. Optionally, entering *,-nnn displays the page beginning nnn lines before the end-of-file.
+ or -	Display the page preceding/succeeding the current page. Optionally, entering +nnn displays the page beginning nnn lines before/after the current page.
S	Display in single screen mode (default).
F	Display in full screen mode.
H	Move the current page pointer to the beginning-of-file.
+H	Display the page beginning at the end-of-group that follows the current page pointer.
-H	Display the page beginning at the end-of-group that precedes the current page pointer.
E or Q	End PAGE and exit.
= string	Search the file from the current page for the first occurrence of the specified string. Print the page beginning where the specified string was found. The words NO MATCH are printed if the string is not found.
Pnnn-mmm	Copy lines nnn through mmm to the print file.

If PAGE does not recognize a command, it responds with:

U/x

x Character that PAGE does not recognize.

PMP

Performance measurement programs (PMP) is a program you can use to return statistics on operating system performance and overhead.

To use PMP, you must have a privileged VSOS user number. PMP uses the privileged F015 resident system call.

Following are the PMP controllee file names and functions:

PMP File Name	Function
READPMP	Generates a report listing both performance data and history code frequency data [F015 subfunction code 6 (110 binary)].
READHG	Generates a report listing only history code frequency data [F015 subfunction code 4 (100 binary)].
READPRF	Generates a report listing only performance data [F015 subfunction code 2 (010 binary)].
CLEARPMP	Clears both the performance data and history code counters [F015 subfunction code 7 (111 binary)].
CLEARHG	Clears the history code counters [F015 subfunction code 5 (101 binary)].
CLEARPRF	Clears the performance data [F015 subfunction code 3 (011 binary)].

To use PMP, follow these steps:

1. Ensure that PERFORM=YES and DISTTIME=YES on the system variables display during VSOS autoload. If DISTTIME=NO, no history code frequency data is returned. When PERFORM=YES is selected, the system allocates an additional two blocks of memory in resident table space for histogram data collection.

2. Either submit a batch job or use an interactive terminal to perform the following steps:

   a. Attach the PMP controllee files to be used (if they are not public files).

   b. Clear the data counters by executing CLEARPMP, CLEARHG, or CLEARPRF.

   c. Run the tasks whose effect on system performance is to be measured.

   d. Generate a report by executing either READPMP or READHG.

PMP REPORT

The PMP report comprises statistic tables on the following measures of system activity:

- Function request processing
- Disk unit activity
- Boat response time
- CPU state
- History code frequency
- Scanner/Pager time distribution

All values are specified in decimal.

Function Request Statistics

The PMP report begins with statistics that break processing time down by system function, as shown in figure 6-7 (report entries are defined following the figure).

F.C.	NUMBER	REALTIME(SEC)	AVERAG (MILSEC)	V.S. CPU(SEC)	AVERAG (MILSEC)	V.S.RT(SEC)	VS AVERAG RT(MSEC)	USER RT-VS RT
0001	246	5.512	22.405	.889	3.612	4.901	19.922	.611
0002	229	4.430	19.345	.790	3.450	3.663	15.996	.767
0003	256	8.831	34.496	.792	3.094	7.694	30.054	1.137
0004	5	.138	27.526	.010	2.034	.137	27.447	.000
0005	227	34.020	149.868	1.038	4.572	33.406	147.163	.614
.	.	.	.	.	.	.	.	.7
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
C502	10851	2.697	.248	.000	.000			
C503	7763	1.984	.255	.000	.000			
C504	99	.014	.145	.000	.000			
	308539	257.572		39.556				38.352

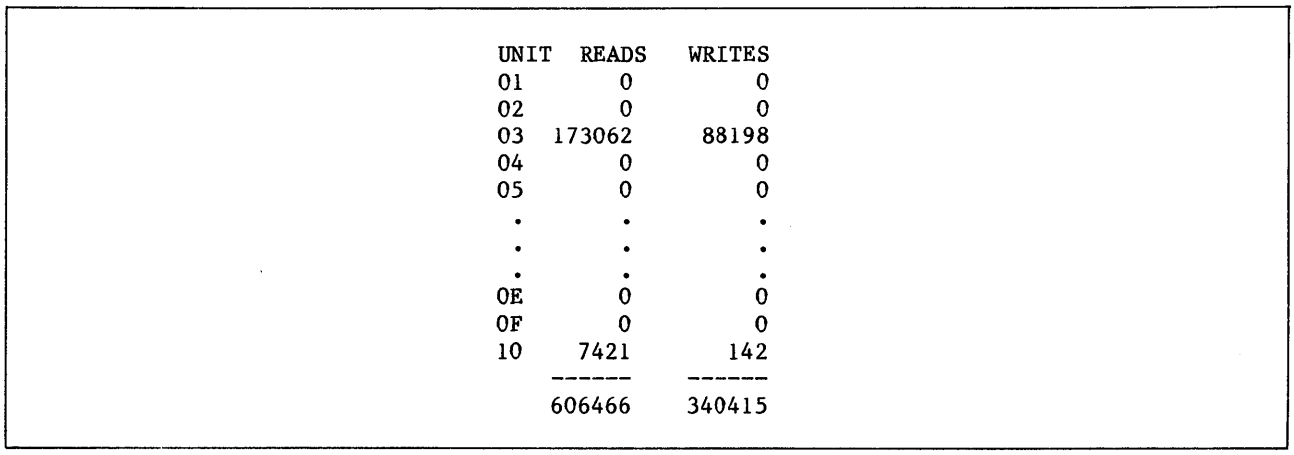
Figure 6-7. Function Request Statistics

Heading	Content
F.C.	Function code. The list begins with the virtual system functions followed by the resident system functions (those that begin with C or F).
NUMBER	Number of requests for the function.
REALTIME (SEC)	Cumulative real time, in seconds, consumed by the function requests. The real time for a function request begins when a user task requests a function, and ends when it receives the function response code.
AVERAG (MILSEC)	Average real time per function request in milliseconds.
V.S. CPU (SEC)	Cumulative virtual system CPU time, in seconds, used by execution of the function requests.
AVERAG (MILSEC)	Average virtual system CPU time per function request in milliseconds.
V.S. RT (SEC)	Cumulative virtual system time, in seconds, used by processing of the function requests. Virtual system time for a function begins when the virtual system takes the function request from the queue, and ends when it completes the function.
VS AV RT (MSEC)	Average virtual system time per function request in milliseconds.
USER RT-VS RT	Cumulative real time minus cumulative virtual system time for the function. The difference is the cumulative time the function requests were queued in seconds.

Virtual system time measures do not apply to the resident system functions.

Disk Unit Statistics

The second table in the PMP report gives disk unit statistics, as shown in figure 6-8 (report entries are defined following the figure).



UNIT	READS	WRITES
01	0	0
02	0	0
03	173062	88198
04	0	0
05	0	0
.	.	.
.	.	.
.	.	.
0E	0	0
0F	0	0
10	7421	142
	606466	340415

Figure 6-8. Disk Unit Statistics

Heading	Content
UNIT	Disk unit number.
READS	Number of 512-word blocks read from the disk unit.
WRITES	Number of 512-word blocks written to the disk unit.

Boat Response Time Statistics

The third part of the PMP report gives boat response time statistics, as shown in figure 6-9 (report entries are defined following the figure).

```
DT=    2159021420
DT2= 476414854456276.000
PGS=        946880
PGS2=     104183418
DT*PGS=    145093743038.000
BTS=         37253
AV.BRT=  58.0
AV.PRT=   2.3
```

Figure 6-9. Boat Response Time Statistics

Heading	Content
DT=	Total boat response times (Σt). A boat response time (t) is the time, in microseconds, from the issuance of the boat to the receipt of the station or NAD response.
DT2=	Total of the squares of each boat response time ($\Sigma +^2$).
PGS=	Total number of blocks the boats transferred (Σ pgs).
PGS2=	Total of the squares of the blocks transferred by each boat (Σpgs^2).
DT*PGS=	Total of each boat response time multiplied by the number of pages the boat transferred (Σ[t * pgs]).
BTS=	Number of boats in decimal.
AV.BRT=	Average boat response time.
AV.PRT=	Average block response time (the cumulative boat response times divided by the total blocks transferred).

CPU State Statistics

The fourth part of the report is a table showing time statistics for each CPU state, as shown in figure 6-10 (report entries are defined following the figure).

```
KERNEL =     184.542      3.076     2.4 %
PAGER  =     137.655      2.294     1.8 %
V.S.   =      53.527       .892      .7 %
USER   =    6827.349    113.789    90.5 %
WAIT   =     300.489      5.008     4.0 %
IDLE   =      41.584       .693      .6 %
TOTAL  =    7545.146    125.752
```

Figure 6-10. CPU State Statistics

The three columns of the table show the cumulative CPU time, in seconds, consumed by the activity, the cumulative CPU time in minutes, and the percentage of the total CPU time spent by each activity, as described below.

Activity	Description
KERNEL	Kernel resident system processing.
PAGER	Pager resident system processing.
V.S.	Virtual system processing.
USER	User task processing.
WAIT	Waiting for completion of I/O requests.
IDLE	In the idle loop.

History Code Frequency Statistics

The history code time distribution table lists the frequency of resident system events, as shown in figure 6-11 (report entries are defined following the figure). Each event corresponds to a history code listed in table 6-2. The table lists history code statistics accumulated since the last VSOS autoload or the last execution of the CLEARPMP or CLEARHG controllees.

The history code mechanism keeps a counter for each history code. When a resident system event corresponding to a history code occurs, time is accumulated on the counter for that code. When another event corresponding to a history code occurs, time is no longer accumulated on the first code counter, but is accumulated on the counter for the next event.

If DISTIME=NO on the VSOS autoload system variables display, no values are returned in the history code time distribution table.

HISTORY CODE TIME DISTRIBUTION

CODE	SECONDS	FREQUENCY	AVG(MILSEC)
001	.015	307	.050
003	1.531	31819	.048
005	1.743	6537	.266
.	.	.	.
.	.	.	.
.	.	.	.
080	.074	1314	.056
0FE	350.437	13975	25.075
0FF	398.093	2864	138.998
	1448821440	376348	
	24.147 MINS		

Figure 6-11. History Code Time Distribution

Heading	Content
CODE	History code. If the event corresponding to a history code does not occur, the code is not listed in the table.
SECONDS	Cumulative seconds recorded on the history code counter.
FREQUENCY	Number of occurrences of the resident system event corresponding to the history code.
AVG(MILSEC)	Average time, in milliseconds, between the occurrence of the resident system event corresponding to this history code and the next event corresponding to a history code.

Table 6-2. History Codes (Sheet 1 of 3)

Code	Event	Routine
1	Job mode illegal instruction.	KERNEL
3	Exit force to KERNEL.	KERNEL
4	Problem program killed.	KERNEL
5	External interrupt.	KERNEL
6	Access interrupt.	KERNEL
7	Virtual process initiated.	KERNEL
8	Exit force to user mode.	KERNEL
13	Periodic task activated.	KERNEL
14	Demand task activated.	KERNEL
15	Call for NAD or KERNEL.	KERNEL
1B	Call to virtual system.	KERNEL
1C	Page deleted from page table.	KERNEL
1D	Sail waiting boat.	SUPPORT/BTISSUE
1E	Virtual system task complete.	KERNEL
20	Zero page fault.	KERNEL
21	Virtual process queued.	SUPPORT
23	Resident periodic task activated.	KERNEL
24	Demand task queued.	KERNEL
25	External interrupt picked up in Scanner.	KERNEL
26	Request from NAD.	KERNEL
27	Response from NAD.	KERNEL NPSRTN

Table 6-2. History Codes (Sheet 2 of 3)

Code	Event	Routine
28	Request from NAD.	RESINIT
29	Enter BOAT in chain.	RESINIT SUPPORT/BTISSUE
2B	C704 call processed.	KERNEL
2C	Page deleted from page table in SUPPORT.	SUPPORT/FF02
2D	NAD request processed by KERNEL.	KERNEL
40	PFAULT called.	NPAGER
41	Advise in.	NPAGER
42	Advise out.	NPAGER
43	Enter page table entry.	NPAGER
44	Enter page table entry for FF01.	SUPPORT/FF01
70	LRU page selected.	NPAGER
71	Purge small pages for large pages.	NPAGER
72	Postpaging called.	NPAGER
73	Postpaging deletion if unmodified page.	NPAGER
74	Postpaging writing of modified pages.	NPAGER
75	WRPLY (response from KERNEL).	NPAGER
76	Last request page table search.	NPAGER
77	Select VS LRU page.	NPAGER
78	Select user LRU page.	NPAGER
79	Postprocessing memory cleanup.	NPAGER
7A	Storage move for large page.	NPAGER
7B	Storage move for large page.	NPAGER

Table 6-2. History Codes (Sheet 3 of 3)

Code	Event	Routine
7C	Storage move one small page.	NPAGER
7D	Write violation page fault.	NPAGER
7E	Extended page table search.	NPAGER
7F	WORKSET called.	NPAGER
80	Evaluate working set routine called.	NPAGER
91	XIOCALL re-entered after page fault.	XIOCALL
92	XIOCALL re-entered after gap pattern.	XIOCALL
93	XIOCALL re-entered after file extension.	XIOCALL
94	File extension error.	XIOCALL
95	File extension requested.	XIOCALL
96	File extension not allowed.	XIOCALL
97	XIOCALL re-entered after buffer page fault.	XIOCALL
98	Only part of buffer used.	XIOCALL
99	Early exit from XIOCALL.	XIOCALL
9A	Explicit I/O.	XIOCALL
9B	Response from NAD.	XIORTN
FE	System wait state.	KERNEL
FF	System idle state.	KERNEL

SCANNER/PAGER TIME DISTRIBUTION

The Scanner/Pager time distribution table lists timing statistics for resident system events, as shown in figure 6-12 (entries are defined following the figure).

For each Scanner/Pager code, the resident system timing mechanism records the number of occurrences for each code and the cumulative time used. The average time for each occurrence is computed from those values.

Table 6-3 lists the Scanner/Pager event codes.

SCANNER/PAGER TIME DISTRIBUTION

CODE	SECONDS	FREQUENCY		AVG(MILSEC)
101	22.120	454272		.048
102	1.871	3999		.467
103	.741	6363		.116
104	45.938	176240		.260
105	1.365	29456		.046
.	.	.		.
.	.	.		.
.	.	.		.
142	3.356	35473		.094
143	182.982	332	551.151	
144	.000	5		.094
	7689925104	3039131		
		128.165 MINS		

Figure 6-12. Scanner/Pager Time Distribution

Heading	Content
CODE	Timing code. If the event corresponding to a code does not occur, the code is not listed in the table.
SECONDS	Cumulative seconds used for all occurrences of this event.
FREQUENCY	Number of occurrences of the event.
AVG(MILSEC)	Average time, in milliseconds, for each occurrence of the event.

Table 6-3. Scanner/Pager Timing Codes (Sheet 1 of 2)

Code	Resident System Event
101	Begin Scanner.
102	Demand task processing.
103	Periodic task processing.
105	Checking for station requests.
106	Process station responses.
107	Exit force instruction to job mode.
108	Begin UPDATE accounting.
109	End UPDATE accounting.
110	Pager called for user page fault.
111	Return from Pager.
112	Pager called for KERNEL page fault.
113	Pager called to simulate write access.
114	Pager called for advise function.
120	KERNEL called by Pager.
121	Return to Pager after request initiated.
122	KERNEL call to Pager after request complete.
123	Return to KERNEL after completion processing.
124	Pager called to request free page.
125	Return to support after free page request.
130	Preprocessing of messages.
131	Message processing complete.

Table 6-3. Scanner/Pager Timing Codes (Sheet 2 of 2)

Code	Resident System Event
135	Station request processing complete.
136	Station response processing complete.
140	Begin wait state.
141	End wait state due to station responses.
142	End wait state due to demand task.
143	Begin idle state.
144	End wait state due to station request.
160	Call RUNIT after message processing complete.
161	Begin postprocessing of message.
162	Virtual system has user keys.

PURGEMCU

Use this utility to allow a privileged user to purge files on the MCU while running under VSOS.

Format:

```
PURGEMCU,name,ID=id,U=unit.
```

name Name of the MCU file required. Required parameter.

id ID of the MCU file. Optional parameter; default is blank.

unit Unit on which the file resides. Required parameter.

QSTATUS (VSOS)

The VSOS QSTATUS Utility periodically gathers information about the VSOS queues, writes the information to a file, and transfers the file (using MFLINK) to each specified and enabled remote host. QSTATUS will then suspend itself for a delay cycle specified on the QSTATUS call. When the delay interval is complete, QSTATUS repeats its cycle of gathering queue information, writing it to a file, transferring the file (using MFLINK) to the remote hosts, etc. An option is provided so that the user may specify the time that QSTATUS begins its cycle of gathering queue information. The QSTATUS information file sent to the remote host may be used by a site-provided remote host utility QSTAT (see QSTAT Utility for remote host in this section) to provide queue information to the user. QSTATUS may only be executed on a privileged user number.

The format of the QSTATUS control statement is as follows:

```
QSTATUS,INPUT=lfn,DELAY=nsecs,TIME=hhmm.
```

lfn Name of the CYBER 200 file which provides the information needed for QSTATUS to transfer its queue information file (using MFLINK) to the remote host. If not specified, the lfn is INPUT. For each remote host, the file must contain the following records:

```
ST=lid{,optional parameters}.
```

lid is the RHF logical identifier of the remote host. The lid should not be shared by other remote hosts. The optional parameters may be the EP, RT, or DD=dd parameters of the MFLINK command and must follow the "ST=lid," character string. Each optional parameter, if specified, is separated by a comma. The last character of the ST=lid record must be a period. This record must precede the JCS record(s) for the remote host.

```
JCS="string1","string2, ...,"stringn.
```

$string_1$ - $string_n$ are the text strings used by MFLINK to send the file to the remote host specified by the previous ST=lid record. The JCS statement may continue on two or more records for up to 185 characters. It is terminated by a "." or ")" not embedded in a string. Strings are not allowed to cross record boundaries.

nsecs The number of seconds that QSTATUS delays before starting its next cycle of gathering queue information and transferring it (using MFLINK) to the remote hosts. If not specified, the value is 300 seconds (5 minutes). The minimum value is 30 seconds. The maximum value is 1800 seconds.

hhmm The time, using a 24-hour clock, that QSTATUS waits before it begins its first cycle of gathering information. If hhmm is after the current time, QSTATUS waits until hhmm. If hhmm is before current time, QSTATUS does not wait. If TIME is not specified, QSTATUS does not wait.

Messages issued by QSTATUS are as follows:

```
I QSTATUS 8339 INPUT FILE IS EMPTY
I QSTATUS 8341 INPUT FILE lfn IS NOT FORMATTED CORRECTLY - PROBLEM
               IN nth RECORD
I QSTATUS 8342 ARGUMENT ERROR FOR SPECIFICATION OF param
I QSTATUS 8343 JCS LONGER THAN 4096 CHARACTERS
I QSTATUS 8346 NO MORE THAN nnnn LIDS CAN BE SPECIFIED
I QSTATUS 8347 USER NUMBER
I QSTATUS 8348 CHECKPOINT CANNOT COMPLETE WHILE QSTATUS IS
               EXECUTING ON SYSTEM USER NUMBER nnnnnn
```

Note that all the above messages are issued as informative messages so that QSTATUS terminates with a return code of zero. This is to allow sites to set up EXIT card processing when normal termination of QSTATUS is performed via an operator DROP, which causes a return code of 8.

The input file for QSTATUS may be built with the help of an editor or by using the BLDIFILE utility described in this chapter. The following is an example of an input file for QSTATUS. The example assumes there are three remote hosts that are front ends to the CYBER 205, whose LID is CY2. The remote hosts have LIDs of M10 (a NOS front-end), MFF (a NOS/BE front-end), and MIB (an IBM front-end). The input file may have the following content:

```
ST=M10.
JCS="USER,12827AA,APASSWD,NOSCLSH","REPLACE,MFSTCY2".
ST=MFF.
JCS="ACCOUNT,7196,7ABCDEF.","PURGE,MFSTCY2,ID=SYSTEM,LC=1.",
"CATALOG,MFSTCY2,ID=SYSTEM.".
ST=MIB.
JCS="REPLACE,DSN=MFSTCY2.QSTAT,UNIT=SYSDA,",
"LRECL=80,RECFM=BBS.".
```

QSTATUS INFORMATION FILE FORMAT

QSTATUS writes its queue information in ASCII into a R type file called Q5ST205. The file contains the following kinds of records in the following order:

```
Date-time record
Keyword record
Bulletin record(s) (optional)
End-bulletin record
Status records
```

The date-time record contains the PID of the CYBER 205, the date and time the file was written, and the delay interval between updates. Its format, described with FORTRAN syntax, is as follows:

```
'STATUS CYBER 205 - ',A3,', DATE: ',A8,' , TIME: ',A8,' PERIOD: ',A4
    A3      PID of the CYBER 205,
    A8      Date in dd/mm/yy,
    A8      Time in hh/mm/ss.
    A4      Period in seconds between updates.
```

The keyword record of the file describes the keywords that the remote host QSTAT may use when extracting information from the status records. The order of the keywords is also the order used when QSTATUS writes the associated data fields within each status record. The list of keyword descriptions are contiguous with a blank (space) character separating each keyword description. Each keyword description has the following format:

```
<keyword><typechar><title>=<width>
```

keyword — The mnemonic that the remote host QSTAT user may use when calling QSTAT.

typechar — A one character variable that the remote host QSTAT uses to determine what is displayed. It may be any of the following characters:

- `:` Keyword is optional, but item is always displayed.
- `-` Keyword can't be used, but item is always displayed.
- `!` Keyword must be used and item will be displayed.
- `#` Keyword is optional, item is only displayed if keyword is selected.
- `*` Keyword must be used and default item is not displayed.
- `/` Keyword must be used, item is hashed and item is never displayed.

title — The full name of the keyword.

width — A numeric string variable which describes the width of the actual value of the keyword within the status records.

The keyword record of the released QSTATUS has the following value:

```
"QU:QUEUE=2 JDN:JDN=4 NAME:NAME=8 UN!USER=6 LID:LID=3 "
"PRI#PRIORITY=6 TASK:TASK=8 STAT:STATUS=12 TL:TL=6 WS:WSET=4 "
"LP:LP=2 CBC#CBC=4 JCAT:JOBCAT=8 PW/PW=16 FST#FILE_STATUS=34"
```

The optional bulletin record(s) follows the keyword record. It may be a general status bulletin about the 205. In the released QSTATUS, there are two bulletin records which contain a summary of the CYBER 205 queues. They have the following format:

```
"  QUEUE:            INPUT   EXEC   OUTPUT"
"  JOBS IN QUEUE:       n      m        p
                  n   The number of jobs in the input queue.
                  m   The number of jobs in the execute queue.
                  p   The number of jobs in the output queue.
```

The end-bulletin record marks the end of the bulletin records. The status records follow the end-bulletin record. The end-bulletin record has the following format:

```
"**END**"
```

The status records contain the actual data values (packed) for the keywords described in the keyword record. The first character of each status record is used as a control character for the remote host QSTAT. If the first character is a *, the status record contains information about a system or system-privileged task. Any other character in the first position indicates that the status record is normal. The remainder of the record contains packed data fields in the same order as the keywords in the keyword record with the width of fields the same as the associated width of the keyword. The following are examples of status records for the input queue, execute queue, system record, and output queue.

```
" IN 410KMT01AFM010801M10 4 129        JMAX          300 512"
    " 3    JDEFAULTZ23HS7V82NAA23BP"
" EX 816JQSTATAN012827M10 4 248QSTATUS RUNNING       4993264"
    " 02816      R7CKKOERGGGGGGGGG"
"*EX 1016SYSTEM  000098   15 255OPERATORW MSG FM CTR 38393264"
    " 04096        HIDX32GGGGGGGGG"
" PR 355PYY395C2010037M10      COUNTAGP                    "
    "              MOTRXZMPTGGGGGGGPRINT"
```

The following is an example of a job which will execute QSTATUS and put itself into the input queue if an operator should drop it during execution.

```
QSTATJOB.
USER,U=qstusernum,PA=password.
RESOURCE,TL=777777,PR=15,JCAT=P15.
ATTACH,QSTATDIR.
QSTATUS,I=QSTATDIR.
EXIT.
COPY,INPUT,QSTATJOB.
PURGE,QSTATJOB.
DEFINE,QSTATJOB.
SUBMIT,QSTATJOB,ST=lid,JCS="route directive".
(eog)
QSTATJOB.
USER,U=qstusernum,PA=password.
RESOURCE,TL=777777,PR=15,JCAT=P15.
ATTACH,QSTATDIR.
QSTATUS,I=QSTATDIR.
EXIT.
ATTACH,QSTATJOB.
SUBMIT,QSTATJOB,ST=lid,JCS="route directive".
```

The QSTATJOB can be altered so that if QSTATUS fails, the QSTATJOB can be sent to a front-end input queue to be sent to the CYBER 205 input queue. It is suggested that sites install a unique job category (JCAT) for the QSTATUS job in order to provide a convenient means to control QSTATUS execution.

QSTAT (REMOTE HOST UTILITY)

The QSTAT utility for the remote host allows the user to obtain information from the status file sent to the remote host by VSOS QSTATUS (see description this section). The remote host QSTAT is not a supported product of CONTROL DATA. However, suggested source code written in FORTRAN 77 for a QSTAT for NOS and NOS/BE is on VSOS's TOOLPL as the decks QSTATNOS and QSTATNBE respectively. Sample build jobs are QSTAT for NOS and NOS/BE are also on TOOLPL as the decks BJQSNOS and BJQSNBE respectively. The suggested code and sample build jobs can be used as a base for adaptation by a site to install a "local code" QSTAT utility for each of their front-ends to VSOS. VAX and IBM remote hosts may use NOS/BE decks as a guide to building their QSTAT utility. Security for the status file will depend on the security features available on the front-end and QSTAT's use of these features. If the VSOS QSTATUS is installed with passwords enabled (see table 5-19), the front-end QSTAT will only allow a user to view queue information about jobs for which the user specifies the user number and password.

The sample QSTAT allows the user to select, via keywords, the kind and amount of information to obtain from the status file. Among the options available are:

```
Information about available keywords (Help prompt)
All files in any or all of the input, execute, and output queues.
All queue files of a particular user number.
Information about a particular job.
Information about jobs from a particular lid.
Information about a task.
```

The following is the format for the call to the sample QSTAT on NOS.

```
QSTAT,lidl,L=lfn,LO=opt,QU=queue,JDN=jdn,NAME=name,UN=usernum,
    LID=lid2,PRI=priority,TASK=task,STAT=status,TL=tl,WS=ws,
    LP=lp,CBC=cbc,JCAT=jcat,PW=pw,FST=fst,HELP
```

lidl — Logical id of the CYBER 205 (optional). The default is specified upon installation of QSTAT.

lfn — File to receive QSTAT output (optional). The default is OUTPUT.

opt — List options (optional). Values may be:

- F — Date/time and status records, including suppressed fields (except passwords are displayed.
- H — Date/time and status records are displayed (default).
- I — Date/time, bulletin, and status records are displayed.
- N — Only status records are displayed.

queue — CYBER 200 queue (optional). Values may be IN, EX, or PR. The default is that all queues are listed.

jdn — Job descriptor number (optional). Values may be 1 to 2047.

name — CYBER 200 job name (optional).

usernum — CYBER 200 user number (requred).

lid2 — Lid from which the job was submitted (optional).

priority	CYBER 200 priority (optional).
task	Name of the task executing (optional).
status	Status of the executing task (optional).
tl	Time limit associated with the job (optional).
ws	Working set block count (optional).
lp	Large page count (optional).
cbc	Current working set count (optional). This item is suppressed unless LO=F is selected.
jcat	Job category (optional).
pw	Password associated with usernum (required). This is never displayed.
fst	Output file status (optional). This item is suppressed unless LO=F is selected.
HELP	Provides information on available keywords (optional).

READMCU/WRITEMCU

Use this utility to read an MCU file onto a CYBER 205 mass storage file or write a CYBER 205 mass storage file onto an MCU file. A user must be privileged to run this utility.

Format:

```
READMCU(I=mcufile,ID=id[,O=sysfile][,T=type][,U=unit])

WRITEMCU([I=sysfile,]O=mcufile,ID=id[,T=type][,U=unit])
```

mcufile	Required MCU file name to be read from or written to.
id	Optional MCU file identifier. Default is two spaces.
sysfile	Optional mass storage file name. If not specified, its default is the same name as the MCU file. For the READMCU utility, a file by this name is created that has the same length as the MCU file. For the WRITEMCU utility, this file must already exist on mass storage. If this file is larger than the MCU file, it is truncated.
type	Optional MCU file type (0=read/write,1=read only). Default is 0.
unit	Optional logical MCU unit (0 through 6 for physical unit 0, 10 through 16 for physical unit 1). If none is specified, MCU searches all logical units in ascending order to look for a matching MCU file name and identifier.

SHORTEN

SHORTEN converts a full memory dump file into a partial memory dump file. The following steps are taken:

1. The full dump image is searched for TABST.
2. The full dump image OPTIONS word in MISCTAB is checked to ensure that it is not actually a partial dump of memory.
3. The full dump image from block 0 through the end of the shared tables is copied to the partial dump image.
4. The partial dump image OPTIONS word is modified to indicate a partial dump.
5. The page table image in the full dump is scanned and pages belonging to the virtual system are copied to the partial dump image.

Format:

```
SHORTEN(FULL=fullfile,PARTIAL=partfile)
```

fullfile	name of the full memory dump file. (default: FULLDUMP)
partfile	name of the partial memory dump file. (default: PARTDUMP)

Dayfile Messages:

```
W FILE filename IS ALREADY PARTIAL DUMP
SHARED REGION ENDS AT #xxxxxxx
MISCTAB AT #xxxxxxx
ARGUMENT ERROR
TABST NOT FOUND
```

SOT

SOT is an interactive, line-oriented text editor.

Format:

```
SOT,inpfn[,O=outfn,p]
```

inpfn — File to be edited. If inpfn does not exist, it is created. Required parameter.

outfn — Optional file to be created to contain the result of the editing session. If outfn is specified, inpfn is not changed as a result of the editing session.

p — Flag that indicates inpfn is a permanent file to be attached. Optional parameter.

Listed next are the commands available using SOT. These are the conventions for the commands:

Convention	Description
L	Used to indicate the last line of the file.
A	Used to indicate a range of lines that encompasses the entire file.
,	Parameter separator. You may also use a space instead of a comma.
;	Default tab character.
@	Any string preceded by @ is assumed to be JCL and will be passed to the operating system.
. or)	Separates commands from parameters. (Commands may be entered on the call line following the file information parameters.)
$	Separates commands from one another.
&	Indicates a blank is to replace the current character.
^	Indicates that the character is to be deleted.

For all commands where a line range is valid, the second line number is optional.

If no strings are given to alter the command, SOT will display the string; then prompt "LINENUMBER = ". The line will be modified as it is entered.

Delimiters for string commands can be any character.

Command	Usage
#A,LN[,/string1/string2/]	Alter LN to replace the first occurrence of string1 with string2. Display the line for modification if string1 and string2 are not given.
#C,L1[:L2],/string1/string2/	Change all occurrences of string1 to string2 in line(s) L1 to L2.
#D,L1[:L2],LN	Duplicate line(s) L1 to L2 to follow LN.
#E	End the edit session; write the updated file to the result file.
#F,L1[:L2],/string/	Find all occurrences of string in line(s) L1 to L2.
#H	Display a summary of all edit commands.
#I,LN	Change the insertion line to LN. If LN is not specified, insert it at the end of the file.
#L,L1[:L2]	List the contents of the file from line(s) L1 to L2. If L1 is not specified, display beginning at the last list command.
#M,L1[:L2],LN	Move line(s) L1 to L2 to follow line LN.
#P,L1[:L2]	Purge line(s) L1 to L2.
#Q	End edit session; do not write the results to the result file.
#R,LN/new line/	Replace line LN with new line.
#S	Write the updated file to the result file; then continue edit session.
#T,chara	Set the tab character to be chara.
#T=C1,C2,C3,C4,C5	Set the tab columns. Default is 1 tab to 7 columns. If there are no parameters given, display tab information.

SPY

Use the SPY routine to compare the relative CPU time spent in each subrange within a virtual address range. Using SPY, you can determine which subranges within your code would benefit most from optimization.

To measure relative CPU time, the SPY routine maintains a counter for each subrange within the virtual address range. At a specified time interval during execution of the virtual address range, SPY reads the current program address and increments the counter for the subrange that includes the address. A call to the SPY routine produces a report containing the accumulated statistics.

To use the SPY routine, follow these steps:

1. Insert SPY routine calls in the source text of the program, including a call to initialize SPY and a call to produce a report.

NOTE

Use of the FORTRAN-supplied function SECOND within the program you want to measure will cause a fatal FORTRAN runtime error. Remove any SECOND function references within your program before using SPY.

2. Compile the source program, then build a controllee file using the program binary file and the SPY routine binary file.

3. Execute the controllee file.

INITIALIZATION CALLS

To use SPY, you must insert a SPY initialization call immediately before the code to be analyzed. The SPY initialization call specifies the parameter values used. The SPY parameters are described in table 6-4. The default value is used if a parameter is omitted or the specified parameter value is zero.

Table 6-4. Parameter Input

Parameter	Value	Default
P1	Starting hexadecimal bit address of sample range.	#8080
P2	Ending hexadecimal bit address of sample range.	Starting address + 10000 in hexadecimal
P3	Subrange (bucket) length in decimal half words.	(Sample range in half words) / (P5 - 2) + 1
P4	Sample time interval in decimal microseconds.	500
P5	Number of subranges in decimal. The maximum number of subranges is 4000.	34 if P3 not specified; otherwise, (sample range in half words) / P3
P6	Print threshold percentage (decimal fraction from 0.000 to 0.999). If the value accumulated for a subrange is less than the threshold percentage of the total sample, SPY does not print the value.	0.000 (all values printed)

Available SPY initialization calls differ in their source of parameter output. SPY initialization calls are listed next.

Initialization Call	Parameter Input Source
SPYINIT	Parameter input is read from the first card image on a file. The file name must be equated to TAPE99 on the PROGRAM statement. The card image must have the following format:

Columns	Parameter
1 through 10	P1
11 through 20	P2
21 through 30	P3
31 through 40	P4
41 through 50	P5
51 through 60	P6

SPYDEF — Parameter input is read from the call itself. The parameter order in the call statement is:

```
CALL SPYDEF(P1,P2,P3,P4,P5,P6)
```

SPYTTY — Parameter input is read from the interactive terminal that initiated controllee execution. (If SPYTTY is used, the SPY routine fails if the controllee is executed in a batch job.)

Figure 6-13 shows the prompt messages sent to the interactive user. If you enter HELP, the prompt messages are repeated.

You must enter the six parameter values separated by commas on one line. If fewer than six values are entered, the SPY routine stops.

```
TYPE IN SPY INPUT PARAMETERS IN THE FOLLOWING FORMAT-P1,P2,P3,P4,P5,P6
    P1= STARTING SPY ADDRESS(HEX BIT ADDRESS)
    P2= ENDING SPY ADDRESS
    P3= BUCKET WIDTH(DECIMAL HALFWORDS)
    P4= TIME INTERVAL/MICRO SECONDS (DECIMAL)
    P5= NUMBER OF BUCKETS (DECIMAL)
    P6= SPY OUTPUT CONTROL (FLOATING DECIMAL)
WAITING FOR SPY INPUT
```

Figure 6-13. SPY Interactive Prompt Messages

REPORT CALLS

A SPY report call generates a SPY report on the OUTPUT file. SPY report calls follow.

Report Call	Action
SPYEXIT	Terminates SPY sampling, stops controllee execution, and generates a SPY report.
REPORT	Generates a SPY report without terminating SPY sampling or controllee execution. The REPORT call has the following format: `CALL REPORT(P7)` The parameter P7 indicates whether the SPY counters should be cleared or their values retained. If the P7 value is zero, the counter values are retained; otherwise, the counters are cleared.

SAMPLING CONTROL CALLS

If you want SPY sampling only during a portion of controllee execution, you can call the SPYOFF and SPYON sampling control calls to disable and reenable SPY sampling.

Sampling Control Call	Action
SPYOFF	Terminates SPY sampling although the controllee continues executing. The SPYOFF call has the following format: `CALL SPYOFF(P8)` The parameter P8 determines whether the call generates a report. If P8 is zero, no report is generated. If P8 is nonzero, the call generates a SPY report.
SPYON	Resumes SPY sampling as the controllee executes.

NOTE

Within the program, a SPY initialization call must precede the SPYON call.

BUILDING THE CONTROLLEE FILE

To build the controllee file using the program binary file and the SPY routine library file, follow these steps:

1. Ask site personnel how to access the SPY routine library file. The site extracts the SPY source text from the TOOLPL library.

2. Build the controllee file using a LOAD control statement that specifies the program binary file and the following LIBRARY parameter:

   ```
   LIBRARY=SPY2LIB
   ```

   Additional LOAD parameters can be specified.

SPY REPORT EXAMPLE

The report generated by the SPY routine consists of a heading followed by a table of statistics. Figure 6-14 is an example of a SPY report (entries are described following the figure). The report heading gives the parameter values that SPY used in the run.

```
SPY OPTIONS SELECTED FOR THIS RUN(EITHER CARD INPUT OR PROGRAM DEFAULT)
      STARTING SPY ADDRESS=0000400000
      ENDING SPY ADDRESS=0000488000
      NUMBER OF BUCKETS= 1000
      BUCKET WIDTH(HALFWORDS)=  18
      SAMPLE INTERVAL(MICRO SECONDS)=  1000
      PRINT SUPPRESS PERCENTAGE=.000


  BEGIN
 P-RANGE     FREQ     %   CUM.%                              HISTOGRAM

0000000000        0    0    .000
0000400000        0    0    .000
0000400240        0    0    .000
0000400480        0    0    .000
    .             .    .      .
    .             .    .      .
    .             .    .      .
0000487900        7    0    .934 *
0000487B40        0    0    .934
0000487D80        0    0    .934
0000487FC0        0    0    .934
0000488020        0    6   1.000 *******

SAMPLE SIZE =     29821
```

Figure 6-14. SPY Report Example

Heading	Content
BEGIN P-RANGE	Starting hexadecimal bit address of a subrange (P-RANGE).
FREQ	Number of times the sampled program address was within the subrange.
%	Percentage of CPU time spent within the subrange.
CUM.%	Cumulative percentage of CPU time spent within the subrange and all previous subranges.
HISTOGRAM	Row of asterisks representing the relative frequency for the subrange.

System SPY is activated by the AUTOCON parameter SPY=YES. The CPU monitor code in KERNEL is controlled by three words in the resident table TABST:

TABST(65) Contains the name of the controllee to monitor. If zero, the virtual system is selected.

The initial value is equal to CNSPY.

TABST(70) Bit zero is the monitor select flag; zero for CPU usage or 1 for virtual page fault analysis.

Bits 1 through 15 contain the CPU sample interval in microseconds.

Bits 16 through 63 contain the starting virtual address to monitor.

The initial value is #012C000000400000.

TABST(73) Bits 0 through 15 contain the width of sample bucket. The width is determined by the formula 2** (value of bits 0 through 15).

Bits 16 through 63 contain the ending virtual address to monitor.

The initial value is #000A000000C00000.

For the preceding value, the bucket width equals 2**A, which is #400 bits.

If the virtual system is being analyzed, bit 16 of the virtual addresses should be set to zero.

For example, setting

```
TABST(65)=0
TABST(70)=#0064400000600000
TABST(73)=#000B400000800000
```

indicates that the virtual system address range #C00000600000 to #C00000800000 is to be used with a CPU sample interval 100 microseconds and a bucket width of 800 bits.

To build the report controllee, use the following load card:

```
LOAD,BINARY,CN=SPYR200/200,CDF=#400,GROS=*A,#420000000000
```

To obtain the SPY report, execute the report controllee. The report is written to the file OUTPUT.

Common block A is loaded to the shared table address.

For SPY data accumulated by KERNEL in the resident operating system, the current table address is #420000000000. The table format is:

16 bits	Sample width
48 bits	Count within the window

Word	Description
1	Flag plus CPU sample interval/SPY beginning address.
2	SPY SHIFT count/SPY ending address.
3	Name of controllee being spied on.
4	Length of SPYDATA array in 64-bit words.
5	Flag to zero or retain current sample counts. If flag is zero, clear the current sample counts to zero. If flag is nonzero, retain the current sample counts.
6-8	Not used.
9	Sample count obtained below the address range.
10	Sample count obtained above the address range.

SYSCOPY

Use SYSCOPY to copy all files from a pool to another pool or to the public set.

Format:

```
SYSCOPY,frm,to(,REPLACE,PACKxx)
```

frm	Pool from which files are to be copied. Required parameter.
to	Pool or 0 indicating to which pool or public set the files are to be copied. Required parameter.
REPLACE	Optional indicator specifying that files in pool frm are to replace existing files.
PACKxx	Optional name of pack on which copied files are to reside.

SYSCOPY must be run by a privileged user if files are to be placed in the public set. Special controllees that must be privileged (MFLINK, CHARGE, and so on) will be made privileged only if SYSCOPY is run by a privileged user.

NOTE

If a file in the frm pool also exists as an attached file, SYSCOPY will use the attached file to copy to and give to the to pool. If the user requires the file in the to pool to have special properties, you should request that before SYSCOPY is run by entering:

```
REQUEST,VSLOD/448,NOSEG,NOEXT.
SYSCOPY.
```

XREF

XREF generates a cross-reference listing of source decks. XREF uses the output file generated by UPDATE as input. The output of XREF is stored on file OUTFILE.

Format:

```
XREF,ulfn,com,olfn/len,OPT=o
```

ulfn Output file from UPDATE. Required parameter.

com Optional parameter that indicates whether common decks are to be processed.

olfn XREF output file; default is OUTFILE.

len Optional output file length.

o Optional parameter to suppress source listing. Specify OPT=N if no source is desired. (N currently is the only valid value for OPT=.)

Example:

```
IN("ACFETCH.RF_DAFJ")
UPDATE,Q,P=RHFPL,C=0,L=9,O=TEMP.
XREF,TEMP.
```

ZAP

Use ZAP to purge and/or return one or more files. Files are selected by file names or prefixes supplied by the user. The user is queried before each file is destroyed if the verify option is selected.

Format:

ZAP(pr1,pr2,...,POOL=poolname,OPTION=option)†

pr1,pr2	ASCII file names or file name prefixes to search (*=ALL). Required parameter.	
poolname	Pool from which files are destroyed. Optional parameter. If the pool is not specified, private files are destroyed.	
option	P	Prefixes (with verify).
	PN	Prefixes (no verify).
	N	No prefixes (no verify).
	Default is no prefixes (with verify).	

† Valid abbreviated entries are underscored.

SECURITY SAFEGUARDS 7

The security safeguards are based on:

- Separation of users into two categories: production and nonproduction user numbers.
- Separation of executable files into two categories: production and nonproduction files.
- Unique job categories for each user category.
- Establishment of a site security administrator user number through which a minimum number of site personnel control the programs that may be run in a production environment.

Several special utilities allow the site security administrator to manage the site and monitor the system to detect possible unauthorized use. The establishment of this security-conscious environment, and the tools to manage it, are described in the following sections.

INITIAL INSTALLATION PROCEDURES

1. Identify the individuals who are to perform the site security administrator function. The number of individuals should be the minimum necessary for adequate coverage during all system operational hours.

2. For an initial installation, establish the site security administrator user number (release value is 999996) as soon as the installation is completed. The release value can be changed to a different number within the operator user number range. Installation parameters IP_ADM_OPRID and IP_ADM_AOPRID must be changed to do this.

   For a system upgrade at an existing site, establish the site security administrator user number before or after installation. In either case, this user number must be privileged and its password given only to individuals authorized to perform the security administrator function.

3. As the virtual system is built, set the installation parameter, IP_DFRSTRT, to one (1). This prevents dropfiles of time limit aborted programs (run by nonproduction users) from being restarted with additional processing time without site security administrator intervention. The site security administrator can enable the restart of such a dropfile by using the special SWITCHP utility. Production users' dropfiles can be restarted without site security administrator intervention.

4. Establish job categories to limit the use of CPU time for nonproduction user numbers.

5. Establish a job category (e.g., PRODUCTN) that only production user numbers are validated to use. This PRODUCTN job category must be the only one that provides unrestrained time limits (maximum time limit and default time limit). All others, including JDEFAULT and INTRACTV, must be set within the following values:

   Maximum time limit (MXTL) - Recommended value = 500 STU/SBU
   Maximum value = 1000 STU/SBU

   Default time limit (DFTL) - Recommended value = 500 STU/SBU
   Maximum value = 500 STU/SBU

These values assume MCHARGE and TCHARGE accounting algorithms equivalent to the release versions.

See chapter 5 for instructions on how to change the job categories of the released system.

BUILDING SPECIAL UTILITIES

The special utilities to be used by the site security administrator are built next. These utilities consist of EDITUDP, PERMITP, SWITCHP, ACC3, and SUM. A build job is included on BUILDPL to build and install these utilities. It is shown in figure 4-1 as XSUTL.

NOTE

The two new PLs, SPSNEW and TOOLNEW, must exist as private permanent files belonging to the user number that is used to run the build job.

EDITUDP provides full user directory editor functions, but must be run by the site security administrator. It can be installed in the system pool, public file set, or given to the site security administrator user number. It is recommeded that the regular EDITUD utility not be built.

PERMITP and SWITCHP do not provide the functions of the regular utilities, but can be installed in the system pool, the public file set, or given to the site security administrator user number. The regular PERMIT and SWITCH utilities should also be installed in the system pool or public file set.

ACC3 and SUM are not restricted to use by the site security administrator. Their use can be controlled by giving them to the site security administrator user number.

A version of the SUM utility is provided on VSOS TOOLPL that runs on a NOS front end system. Called SUM170, it has the same calling parameter definitions as SUM. However, the ACC3 output file, TAPE3, must be MFLINKed to the NOS system for input to SUM170.

The source code for SUM170 is in deck SUM170 on TOOLPL and must also be MFLINKed to the NOS system. A PROC to compile this source is also provided on TOOLPL as deck BSUM170. It expects the source code to be in the indirect, permanent file called SSUM170 and produces the direct, permanent, executable file SUM170. This PROC file must also be MFLINKed to the NOS system.

The site security administrator should choose the numbers to be used as production user numbers and, after system installation, use the special EDITUDP utility to establish them as production user numbers in the user directory.

The site security administrator should review the system utilities built from the release materials for the production user numbers, then construct a batch job to use the special PERMITP utility to give these utilities production status. This job should be executed after system installation. BUILDPL has a representative PROC (SETPROD) that can be used in a job as follows:

```
bldjob,STlid.
USER,U=site security administrator user number,AC=account number,PA=password.
RESOURCE,TL=1000.
IN,SETPROD.
UPDATE,Q,D,8,P=BLDPLX,C=BLDTEMP,L=0.
BEGIN,,BLDTEMP,IPOOL.
```

During site preparation, take steps to control access to the operator terminal and the terminal used by the site security administrator. The regular EDITUD utility should not be installed on the system. If it is, it must be owned by the site security administrator with no other user given access. This ensures that the site security administrator has sole control over the assignment of production status to users and files.

The site security administrator (or a designee) should review the source code of the site's production programs to establish that each performs its intended function. An executable controllee file should then be generated for each authorized production program. The site security administrator must then use the special PERMITP utility to establish each such file as a production controllee. These files can now be executed by the production user numbers.

Production user numbers are intended to execute only programs whose source code has been reviewed. Therefore, while such programs are under development or in checkout, they are run under nonproduction user numbers. If problems arise with a production program, a copy of the executable file is to be given to a nonproduction user number for debugging. Consequently, neither the compliers nor the DEBUG utility should be given production status.

The special utilities provided for the site security administrator's use to establish production user numbers, production files, and to enable the restart of dropfiles are described in the following paragraphs.

EDITUDP

The EDITUDP utility allows the site security administrator to establish production user numbers and user number permission for interactive access to the system.

This utility supports all of the functions of the EDITUD utility described in the Operator's Guide, but has one additional parameter on the ADD and MODIFY directives:

```
ADD/MODIFY (...PRODUCTN=P,...)
```

PRODUCTN=p

p=y establishes the user number as a production user number.

p=n removes production status from the user number.

If the PRODUCTN=p parameter is specified, the utility must be executed under the site security administrator user number.

If the LIST directive is specified, the output includes the interactive access indicator and the production user flag. (An asterisk is appended to the user number if it is a production user number.)

For example:

```
 U       PRIO  PRIV  SEC  VRA  UPC  TAPES  INTR  TIME     FLMAX      AC
               JCAT

111111*  N     N     1    N    N    Y      Y     410331         0   9999999
         P4          P6
```

If LO=I is specified, the CRT display appears as in the following example:

```
 U       PRIO  PRIV  SEC  VRA  UPC  NT  INT  TIME    FLMAX      AC      JCAT
111111*   N     N     1    N    N    Y   Y   410331         0  9999999  P4
                                                                       P6
```

Only one of the user directory editor utilities, EDITUD or EDITUDP, can be active at any one time. If one is active when the other is started, the started one waits until the active one is done. An informative message is output approximately once every minute of waiting:

```
I 28 WAITING FOR ACCESS TO USER DIRECTORY.
```

If the first does not finish within five minutes, the second aborts with the following message:

```
F 29 TIME OUT ON WAIT FOR USER DIRECTORY.
```

This situation is not likely to occur if only the EDITUDP utility is installed under the site security administrator user number.

PERMITP

The PERMITP utility allows the site security administrator user number to indicate that a controllee is verified and can run in a production environment.

It also allows the production status of a controllee to be changed for a public, pool, or private permanent file.

This utility can be used only by the site security administrator user number and has the following control statement format:

```
PERMITP, { lfn, OWNER=ownun                  }, PRODUCTN=p
         { POOL=poolname, lfn                }
         { PUBLIC=lfn                        }
```

lfn,OWNER = ownun	Private permanent file name, lfn, owned by user number, ownun. If OWNER = is not specified, the site administrator is assumed to be the owner.
POOL = poolname, lfn	Poolname and pool file name.
PUBLIC = lfn	Public file name.
PRODUCTN = p	Production status of the controllee. PRODUCTN = Y gives the controllee production status and removes all write access permissions (W,A,M) from the file. (This applies to the owner, the general access permissions, and all other individual access permissions.) PRODUCTN = N removes production status from a file. Access permissions are not changed. If PRODUCTN = i is not specified, production status and access permissions are not changed.

The following fatal error messages may be returned:

1- 99	ILLEGAL PARAMETER
250	REQUIRED PARAMETER MISSING
301	POOL poolname DOES NOT EXIST OR IS NOT ATTACHED
310	CALLER NOT PRIVILEGED
517	CALLER IS NOT SITE ADMINISTRATOR
1402	FILE filename DOES NOT EXIST
1682	UNDEFINED USER NUMBER
1704	NO ROOM FOR USER TABLE, FILEI, OR FST ENTRIES

SWITCHP

The SWITCHP utility allows the site security administrator user number to grant a development user number permission to restart a dropfile of a program that aborted when the time limit was exceeded. (Dropfiles owned by production user numbers can be restarted without intervention.)

The control statement format is as follows:

```
SWITCHP,lfn,OWNER=ownun, RESTART=y.
```

lfn	Private permanent file name of the dropfile.
OWNER=ownun	Owner's user number. If OWNER = is not specified, the site administrator is assumed to be the owner.
RESTART=y	Enable restart of dropfile.

The following fatal error messages may be returned:

1- 99	ILLEGAL PARAMETER
250	REQUIRED PARAMETER MISSING
303	INVALID FILE NAME
310	CALLER NOT PRIVILEGED
517	CALLER IS NOT SITE ADMINISTRATOR
519	FILE IS NOT A DROPFILE
1402	FILE filename DOES NOT EXIST
1682	UNDEFINED USER NUMBER
1704	NO ROOM FOR USER TABLE, FILEI, OR FST ENTRIES

PROCEDURAL CONSIDERATIONS

- Do not keep the system source code program libraries (PLs) on line, except when making authorized system modifications, and then only under the direct control of the security administrator.
- Remove the PLs from online storage after the executable files with modifications are prepared.
- Do not give compilers and the LOAD and DEBUG utilities production status. This restriction prevents authorized executable code from being modified under production user numbers.
- Develop programs only under the constraints imposed on nonproduction user numbers.

OPERATIONAL CONSIDERATIONS

- In general, production user numbers, production tasks, and production files are appended with an asterisk when appearing in operator displays or system utilities output.
- The user directory must be modified under direct control of the site security administrator.
- Production files must be archived by the site security administrator if the files are to retain their production status upon reloading using LOADPF. Their production status is lost if reloaded by either another privileged user number or the owner's user number.
- Interactive access can be controlled on a user-by-user basis with the EDITUD or EDITUDP utility.

MONITORING SYSTEM USE

Two utilities, ACC3 and SUM, in combination, enable the site security administrator to monitor system use on a per-user basis, per-job basis, or per-task basis. The ACC3 utility, which produces a printable output file, serves as an intermediate processing step for the system usage monitoring utility, SUM.

ACC3

The ACC3 utility processes a set of system accounting files and generates a new printable output file, TAPE3, which contains:

- Job name.
- User number.
- Task name.
- Job start time.
- Job end time.
- Task start time.
- Task end time.
- Average CPU use.
- Average working set.
- Number of disk I/O requests.
- Number of disk sectors transferred.
- Number of tape I/O requests.
- Number of tape kilobytes transferred for each job and for each task.

An example of file TAPE3 follows:

```
JOB NAME    USER-NO    TASK          START-DATE/TIME      END-TIME
JQSTATWN    12827*     BATCHPRO      06/04/87 20:26:38    20:32:27
                       ATTACH*       06/04/87 20:26:40    20:26:40
                       TOTAL         06/04/87 20:32:25    20:32:26

CY18BKWQ    999998     BATCHPRO      06/04/87 20:31:07    20:32:43
                       PATTACH       06/04/87 20:31:11    20:31:11
                       TOTAL         06/04/87 20:32:37    20:32:37

   CPU       MEMORY        DIO        SECTORS      TIO        KBYTE
 0.217      290.000         9.           30.        0.           0.
 0.028       51.000         0.           0.         0.           0.
 1.863      290.854       104.          431.        0.           0.

 0.262      119.00          9.           30.        0.           0.
 0.027       51.000         0.           0.         0.           0.
 0.948      119.248        55.          231.        0.           0.
```

where:

CPU	= average user CPU use
MEMORY	= average working set
DIO	= number of disk I/O requests
SECTORS	= number of disk sectors transferred
TIO	= number of tape I/O requests
KBYTES	= number of tape kilobytes transferred

If a task name is followed by an asterisk (*), it is a production task. If a user number is followed by *, it is a production user number.

The parameters and directives are described in chapter 6, Tools.

SUM

The SUM utility processes the ACC3 output file, TAPE3, and summarizes resource use on a per-user, per-job, or per-task basis.

SUM can also be used to extract users, jobs or tasks with certain characteristics. For example, it can list all users, jobs, or tasks that have average CPU use greater than 50 seconds.

The control statement format for SUM is as follows:

```
SUM,INPUT=ilfn,LIST=olfn,USERS=u,JOBS=j,TASKS=t,LO=l,
    LCPU=n1, UCPU=n2, LMEM=n3, UMEM=n4,
    LDIO=n5, UDIO=n6, LSEC=n7, USEC=n8,
    LTIO=n9, UTIO=n10, LBYT=n11, UBYT=n12.
```

where:

INPUT=Input file name. Default is INPUT.
LIST=Output file name. Default is OUTPUT.

USERS	Y=	Provides resource use that satisfies the minimum and maximum parameters on a per-user basis.
	N=	Does not provide resource use on a per-user basis.
	*=	Provides resource use on a per-user basis (regardless of the minimum and maximum parameters, which may apply to JOBS or TASKS parameters).
		Default is Y if none of the USERS, JOBS, and TASKS parameters are specified. Otherwise default is N.
JOBS	Y=	Provides resource use that satisfies the minimum and maximum parameters on a per-job basis.
	N=	Does not provide resource use on a per-job basis.
	*=	Provides resource use on a per-job basis (regardless of the minimum and maximum parameters).
		Default is N.
TASKS	Y=	Provides resource use that satisfies the minimum and maximum parameters on a per-task basis.
	N=	Does not provide resource use on a per-task basis.
	*=	Provides resource use on a per-task basis (regardless of the minimum and maximum parameters).
		Default is N.
LO	A=	Only the users (jobs or tasks) that satisfy all the minimum and maximum parameters are listed.
	O=	Users (jobs or tasks) that satisfy any of the minimum and maximum parameters are listed.
		Default is A.
LCPU	=	Minimum CPU time. Default is 0.
UCPU	=	Maximum CPU time. Default is all users/jobs/tasks with CPU time greater than LCPU.

LMEM = Minimum average working set. Default is 0.

UMEM = Maximum average working set. Default is all users/jobs/tasks with average working set greater than LMEM.

LDIO = Minimum disk I/O requests. Default is 0.

UDIO = Maximum disk I/O requests. Default is all users/jobs/tasks with number of disk I/O requests greater than LDIO.

LSEC = Minimum disk sectors transferred. Default is 0.

USEC = Maximum disk sectors transferred. Default is all users/jobs/tasks with number of disk sectors transferred greater than LSEC.

LTIO = Minimum tape I/O requests. Default is 0.

UTIO = Maximum tape I/O requests. Default is all users/jobs/tasks with number of tape I/O requests greater than LTIO.

LBYT = Minimum tape bytes transferred. Default is 0.

UBYT = Maximum tape bytes transferred. Default is all users/jobs/tasks with number of tape bytes transferred greater than LBYT.

If none of the minimum or maximum values are specified, the output related to all of the users, jobs or tasks is listed. If neither minimum nor maximum values are specified for a certain time (for example, Disk I/O requests), this item is not used to determine the output list.

For example:

SUM.	Provides resources use on a per-user basis.
SUM,LDIO=50.	Lists all users that have more than 50 disk I/O requests.
SUM,LDIO=50,UDIO=100.	Lists all users that have more than 50 disk I/O requests and less than 100 disk I/O requests.
SUM,UDIO=100.	Lists all users that have from 0 to 100 disk I/O requests.
SUM,JOBS=Y,UDIO=50.	Lists all jobs that have more than 50 disk I/O requests.
SUM,TASKS=Y,UDIO=50.	Lists all tasks that have more than 50 disk I/O requests.
SUM,U=Y,J=Y,T=Y.	Lists all resource use on a per-user, per-job, and per-task basis.

The format of the user's resource use report is as follows:

USER-NO	DATE	CPU	MEMORY	DIO	SECTORS	DIO	K-BYTES
99	06/04/87	.094	51.000	7.	7.	0.	0.
9151*	06/04/87	3.139	825.600	279.	438.	0.	0.

The user report is sorted by user number.

The format of SUM job list output follows. The example given shows the job list output generated by SUM,JOBS=Y,LDIO=50.

JOB-NAME	USER-NO	START-DATE/TIME	END-TIME
INTRACTV	99	06/14/87 13:49:39	13:51:38
JOBAA	1005*	06/14/87 13:51:35	14:00:20

CPU	MEMORY	DIO	SECTORS	TIO	KBYTES
0.55	102.1	59.0	833.0	0.	0.
14.11	109.5	67.0	845.0	0.	0.

Note that only jobs with DIO greater than 50 are listed.

The format of SUM task list output is as follows:

JOB-NAME	USER-NO	TASK	START-DATE/TIME	END-TIME
PDMPAAWA	999998	BATCHPRO	06/04/87 19:35:05	19:56:24
		ATTACH*	06/04/87 19:35:09	19:35:11
SYSLIBWH	12107*	BATCHPRO	06/04/87 20:02:11	20:02:51
		ATTACH*	06/04/87 20:02:14	20:02:15

CPU	MEMORY	DIO	SECTORS	TIO	KBYTES
0.2462	375.0	9.000	30.00	0.	0.
0.1196	333.0	1.000	2.00	0.	0.
0.5588	314.0	9.000	30.00	0.	0.
0.1197	133.0	1.000	2.00	0.	0.

If a task name is followed by *, it is a production task. If a user number is followed by *, it is a production user number.

The following error messages are returned by SUM:

ILLEGAL PARAMETER: ill_param.	(where ill_param is the illegal parameter.) This is a fatal error. Program aborts.
ILLEGAL VALUE IN PARAMETER:param	(where param is the parameter with an illegal value.) This is a warning error. Program ignores this parameter and continues execution using the parameter default value.
FILE NOT FOUND: filename	(where filename is the name of the file not found.) This is a fatal error.

REMOTE HOST FACILITY (RHF) INSTALLATION NOTES A

INSTALLATION PROCESS

Check the RHF sections of chapter 5 for installation that may need changing. Configuration (or reconfiguration) of RHF involves the following steps (see the VSOS 2 Operator's Guide for detailed descriptions of various operator commands):

1. Add the RCD NAD(s) to the system configuration table (SCT) using the NAD command from the SYST utility (see Initialize the System Configuration Table (SCT) in chapter 2 of this manual). For sites upgrading from 2.3 or 2.3.5, a snap of the previous system's C display is useful when entering the NAD commands. SHD NAD(s) need only be entered if the capabilities of loading, dumping, and error logging of the SHD NAD(s) from the CYBER 205 MCU are desired.

2. Using the CONR command from the SYST utility, define the communication links between each RCD NAD and the SHD NAD(s) that it can communicate with (see Initialize the System Configuration Table (chapter 2) of this manual and the description of CONR in chapter 2 of the VSOS 2 Operator's Guide). For sites upgrading from 2.3 or 2.3.5, a snap of the previous system's C,R display is a useful reference.

3. During initialization of AUTOCON parameters, use the APPL command to define the VSOS RHF applications and the associated remote applications (see VSOS 2 Operator's Guide, chapter 2). For sites upgrading from 2.3 or 2.3.5, a snap of the previous system's N display is a useful reference. Note that there may be up to three APPL entries (VSOS servicers QTFS, PTFS, and SCFS) for the local host, and there may be up to three APPL entries for each remote host (VSOS initiators QTF, PTF, and DLF). The VSOS servicer, SCFS, is required only if IP_SCF is equal to one (refer to table 5-19). There is no actual utility SCFS as PTFS performs the queue status operation, but SCFS is added to provide control over the status functions to keep it separate from the permanent file functions that PTFS also performs. The VSOS initiator, DLF, is required only for those remote hosts that have a front-end archive process.

NOTE

The entries for VSOS servicer applications (QTFS, PTFS, and SCFS) must precede the VSOS initiator applications (PTF, QTF, DLF). If the applications are specified out of order, the front-end will not be able to communicate with the CYBER 205. As front ends attempt to communicate, the T_RHFT (incoming connect) table will accumulate unprocessed incoming connects.

4. During initialization of AUTOCON parameters, use the RHFID command to define the physical identifier (PID) and the associated logical identifier (LID) of each host in the network (see chapter 2 of VSOS 2 Operator's Guide for a description of the RHFID command). For each PID, there must be at least one RHFID command entry where the LID matches the PID. For sites upgrading from 2.3 or 2.3.5, a snap of the previous system's W display is a useful reference.

NOTE

There must be a local RHFID entry for each unique LID used by the remote hosts to designate the CYBER 205. Also, all the local RHFID entries must precede the remote RHFID entries.

5. For VAX and IBM front-ends, if installing remote host utility QSTAT (see chapter 6), use deck QSTATNBE on TOOLPL as the base for developing a local code QSTAT utility. Also see deck BJQSNBE for a sample build job.

INSTALLING LARGE CONFIGURATIONS

The RHF default system shared table space may not be sufficient for larger site networks. The default values allow for networks with up to 14 PIDs and 64 LIDs. Refer to chapter 5 of this manual for a description of the installation parameters to adjust if you need a larger network configuration.

OUTPUT FILE MONITORING

Output files can be monitored with the H,O operator command. Output files are owned by User-6. If a file stays in the H,O display, check the status field for that file in the display. The status entries are self-explanatory and are documented in the VSOS 2 Operator's Guide. Also check the K display for messages. If the LCN network is not functioning, check the RCD and SHD NADs (and front-end RHF) to see if they are running. Also check tables T_CAT and T_RHFT to see if they are filled.

NOTES ON USING RHF

Following is general information on RHF that you may find useful as you install or upgrade your system.

- Analysts may log in to the RHF user numbers. This allows debug support of RHF applications. See the VSOS 2 IMS, Volume V, for a description of how to obtain traces of RHF applications. It is strongly recommended that sites install passwords for user numbers 6, 8, 10, and 12 through 15.

- The system dayfile contains a large amount of RHF history information and can be examined for error messages. A utility is available to analyze the system dayfile for RHF error conditions. The utility is called ASDF and is documented in chapter 6 of this manual.

- If, after installation and/or reconfiguration, VSOS autoloads and runs successfully, but no batch jobs are arriving at VSOS, or front-end MFLINKs to VSOS appear to hang, it is likely that RHF was configured incorrectly.

  Entries in the T_RHFT (incoming connect) table that are not being processed may also signal configuration problems. Check through the configuration process at the beginning of this chapter; be sure to read the two notes in that section.

- Validation of the CYBER 200 LID is being done by the RHF servicer applications (QTFS, PTFS). This means that whatever LIDs the front-end RHF uses to designate the CYBER 200 mainframe must be configured as local CYBER 200 LIDs on VSOS.

SAMPLE CONFIGURATION

Following is a sample VSOS RHF configuration. Suppose there are two front-end machines attached to a single CYBER 205 and the front ends use LIDs TY0, TY1, and TY2 to designate the CYBER 205. Assume that the PIDs (physical IDs), LIDs, and NAD numbers are:

Machine	PID	LID	NAD
CYBER 205	TY0	TY0, TY1, TY2	50 (RCD)
Front-end 1	M21	M21	30 (SHD)
Front-end 2	MFG	MFG	31 (SHD)

The following are the AUTOCON commands used to set up the RHF mainframe table (T_RHMFT):

```
RHFID,TY0,TY0,ON,50,0,LOCAL
RHFID,TY0,TY1,ON,50,0,LOCAL
RHFID,TY0,TY2,ON,50,0,LOCAL
RHFID,M21,M21,ON,30,0,REMOTE
RHFID,MFG,MFG,ON,31,0,REMOTE
```

Note that the RHFID entries that define local CYBER 205 PID/LID entries must precede the REMOTE RHFID entries. Note also that there must be a local RHFID entry for each LID used to designate the CYBER 205.

The following are the AUTOCON commands used to set up the application table (T_APPL):

```
APPL,QTFS,QTF,pw,TY0,n
APPL,PTFS,PTF,pw,TY0,n
APPL,SCFS,PTF,pw,TY0,n (required only if IP_SCF = 1)
APPL,QTF,QTFS,pw,M21,n
APPL,PTF,PTFS,pw,M21,n
APPL,QTF,QTFS,pw,MFG,n
APPL,PTF,PTFS,pw,MFG,n
```

pw The servicers password (if used).

n The maximum number of concurrent operations.

NOTE

The APPL entries that define CYBER 200 servicers (QTFS and PTFS) must precede the APPL entries that define CYBER 200 initiators (QTF, PTF, and DLF).

The N display for this example would appear as follows (if n = 3 and pw = PASSWRD):

Name	Remote Name	Active	Limits	Password	Remote PID
QTFS	QTF		8/3	PASSWRD	TY0
PTFS	PTF		8/3	PASSWRD	TY0
SCFS	PTF		8/3	PASSWRD	TY0 (if APPL for SCFS was entered at AUTOCON time)
QTF	QTFS		8/3	PASSWRD	M21
PTF	PTFS		8/3	PASSWRD	M21
QTF	QTFS		8/3	PASSWRD	MFG
PTF	PTFS		8/3	PASSWRD	MFG

If the site uses the front-end dump and load process, an application table entry for DLF will also be required. If the site has remote systems for NOS/BE or Scope 2, there should be application table entries for those special PTF names. For the sample configurations, these would appear as follows:

```
APPL,PTFS,FTF,pw,TY0,n
APPL,PTFS,PTFU,pw,TY0,n
```

pw The servicer's password (if used).

n The maximum number of concurrent operations.

SCHEDULE INSTALLATION NOTES B

SCHEDULE is a periodic, privileged system task that bridges the communication gap between the input queue scheduler and the executing queue scheduler.

SCHEDULE functions best in an environment where job category definitions follow the general principle of high priority, low resources; low priority, high resources.

Since SCHEDULE does not suspend jobs running under a privileged user number, use discretion when assigning a user privileged status.

SCHEDULE can be tuned to fit a site's needs by changing the value of the variables that are defined in parameter statements at the beginning of the program. These variables define various thresholds for scheduling and for special end case values.

Variable	Default	Location	Description
NO_SUSP	10	UTSCHED.38 SENDMOP.9 BINASCF.7	Maximum number of jobs that can be suspended at one time.
PR_LD	4	UTSCHED.31	Minimum priority of a job in MXMO which can cause lower priority jobs in execution to be suspended. Jobs equal to or below this priority will not cause jobs to be suspended.
PR_HI	11	UTSCHED.32	Maximum priority of a job to be considered for suspension. Jobs equal to or above this priority will not be suspended.
TL_MIN	100	UTSCHED.33	Minimum time limit in STUs or SBUs of a job to be considered for suspension. Jobs equal to or below this limit are not suspended.
WS_MIN	300	UTSCHED.34	Minimum working set of a job to be considered for suspension. Jobs equal to or below this limit will not be suspended.
MEM_MIN	15	UTSCHED.35	Minimum percentage of uncommitted memory necessary before suspended jobs are resumed.

INDEX

Comments (continued from other side)

Please fold on dotted line;
seal edges with tape only.

FOLD

FOLD

FOLD

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
First-Class Mail Permit No. 8241 Minneapolis, MN

POSTAGE WILL BE PAID BY ADDRESSEE

CONTROL DATA
Technical Publications
ARH219
4201 N. Lexington Avenue
Arden Hills, MN 55126-9983

COMMENT SHEET

MANUAL TITLE: CDC VSOS Version 2 Installation Handbook

PUBLICATION NO.: 60459440 **REVISION:** J

NAME: _____

COMPANY: _____

STREET ADDRESS: _____

CITY: _____ **STATE:** _____ **ZIP CODE:** _____

This form is not intended to be used as an order blank. Control Data Corporation welcomes your evaluation of this manual. Please indicate any errors, suggested additions or deletions, or general comments below (please include page number references).

☐ Please Reply ☐ No Reply Necessary

NO POSTAGE STAMP NECESSARY IF MAILED IN U.S.A.

FOLD ON DOTTED LINES AND TAPE

CORPORATE HEADQUARTERS P.O. BOX 0 MINNEAPOLIS, MINNESOTA 55440