

Type III Class A Program

IBM

**Control Program-67/Cambridge Monitor System
(CP-67/CMS) Version 3
Program Number 360D-05.2.005
Installation Guide**

The purpose of this document is to provide the installation with instructions on creating a runnable CP-67/CMS system tailored to his configuration. There are also considerations for tuning the system as well as maintaining both CP-67 and CMS.

This manual should be read in its entirety before installing the system, as procedures have changed.

PREFACE

The following publications are referenced in this manual:

CP-67/CMS User's Guide GH20-0859-0

CP-67 Operator's Guide GH20-0856-0

The following publications provide additional information on CP-67/CMS:

CP-67/CMS System Description Manual GH20-0802-1

CP-67 Program Logic Manual GY20-0590-0

CMS Program Logic Manual GY20-0591-0

CP-67/CMS Installation Guide GH20-0857-0

SCRIPT User's Manual GH20-0860-0

CP-67/CMS Hardware Maintainability Guide GH20-0858-0

First Edition (October 1970)

This Type III Program performs functions that may be fundamental to the operation and maintenance of a system.

It has not been subjected to formal test by IBM.

Until the program is reclassified, IBM will provide for it: (a) Central Programming Service, including design error correction and automatic distribution of corrections; and (b) FE Programming Service, including design error verification, APAR documentation and submission, and application of Program Temporary Fixes or development of an emergency bypass when required. IBM does not guarantee service results or represent or warrant that all errors will be corrected.

You are expected to make the final evaluation as to the usefulness of this program in your own environment.

THE FOREGOING IS IN LIEU OF ALL WARRANTIES EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

This edition applies to Version 3, Modification Level 0, of Control Program-67/Cambridge Monitor System (360D-05.2.005) and to all subsequent versions and modifications until otherwise indicated in new editions or Technical Newsletters.

Changes are continually made to the information herein. Therefore, before using this publication, consult the latest System/360 SRL Newsletter (GN20-0160) for the editions that are applicable and current.

Copies of this and other IBM publications can be obtained through IBM branch offices.

A form has been provided at the back of this publication for readers' comments. If this form has been removed, address comments to: IBM Corporation, Technical Publications Department, 112 East Post Road, White Plains, New York 10601.

CONTENTS

Introduction	1
Installing CP-67	2
CMS for CP-67 SYSGEN	2
Setup of Machine Configuration	4
Selection of Options	5
Real Timer--&RTIMR	6
Virtual 67--&V67	6
ISAM Option--&ISAM	7
Obtaining CP-67	7
Obtaining the Utilities	8
Formatting of Volumes	9
Allocation and Directory Creation	10
Creation of Permanent Residence Volume	11
Obtaining CP-67 from an Existing CMS	12
CP-67 EXEC Procedures	13
CPUPASM	13
CPMACGEN	13
CPMACADD	13
CPMACREP	13
CPGEN	14
CPUPMAC	14
CPSYS	14
CPPUNCH	15
CPLIST	15
CPASMGEN	15
CPTGEN	15
CP-67 Deck Formats	17
CP-67 System	17
CP Utilities	18
Machine Configuration Definition for CP-67	19
REALIO Macro	19

SYSRES Macro	20
SYSGEN Macro	20
SIMPLEX Macro	20
I/O Block Definition Macros	21
Define Channel Macro	21
Define Control Unit Macro	21
Define Device Macro	22
Define Multiplexor Device Macro	22
Generating "Named" Operating Systems under CP-67	25
CP-67 Maintenance Procedure	27
CPPTF EXEC	27
Tuning CP-67	28
Installing CMS	30
DUMP/RESTORE Utility	30
CMS System Disk	32
Block Information	32
Bare Machine--CMS	33
CMS Source Disk	33
Altering the CMS System Disk	34
Nucleus	34
IPL'able System Disk	34
Saving CMS under CP-67	35
CMS EXEC Procedures	36
Maintaining the CMS System Disk	38
CMS Maintenance Procedures	38
Changing Nucleus Resident Commands	39
Changing Disk-Resident Commands	40
C-Disk as a Read-Only Extension of S-Disk	40
CMS Considerations	42
Formatting the P-Disk	42
Obtaining a Nucleus Load Map	42
P-Disk Considerations for the Bare Machine	43
Command Abbreviations	43
Installation of the CMS Batch Monitor	44
Sample Problem	45

INTRODUCTION

The initial installation of CP-67 involves running CMS on the bare machine. CMS is used to load the CP-67 basic distribution tape onto a disk, read in the REALIO configuration for the installation and assemble the REALIO and other modules, if necessary, to select the desired options. Once this is done, an EXEC procedure is invoked to punch the CP-67 nucleus, utilities, and loaders. The punched decks are then used on the bare machine to format the CP-67 volumes (sysres, drums, etc.) and to load the nucleus and directory on the residence volume. If CP-67 is already installed, regenerations or new installations can be performed from a CMS virtual machine.

Version 3 Level 0 of CP-67 has many new features that make it incompatible with previous versions. For this reason, users must not use utilities, loaders, or nucleus decks from previous releases. In particular, the real machine configuration description requires additional information to produce a valid RIO module.

INSTALLING CP-67

CMS FOR CP-67 SYSGEN

To obtain CP-67, the CMS system is used. A description follows of how to obtain CMS and run it on the real machine (not under CP). This description is intended as a convenient summary, as more detailed information can be found under "Installing CMS".

First, restore the DUMP/RESTORE copy of the CMS System disk. The CMS System tape contains the following files:

file 1--IPL'able DUMP/RESTORE
file 2--a D/R copy of the CMS System disk.

Then, IPL the restored disk. Hit REQUEST on the 1052 console. CMS now allows you to reconfigure the CMS device address table to match the real device configuration of the installation. The following question is asked:

Q1: REDEFINE ADDRESSES? (YES,NO):
Answer YES, if the real addresses are different from the standard CMS virtual addresses.

It is then necessary to enter the four-digit addresses for the following devices: console, S-disk, P-disk, reader, punch, printer, tape one, and tape two. Answer NO to the CP-SAVE question.

The standard CMS device addresses are listed below.

<u>Device</u>	<u>Virtual Address</u>	<u>Symbolic Name</u>	
1052	009	CON1	console
2311,2314	190	DSK1	system disk (read-only)
2311,2314	191**	DSK2	permanent disk (user files)
*2311,2314	192**	DSK3	temporary disk (workspace)
*2311,2314	000**	DSK4	A disk (user files)
*2311,2314	000**	DSK5	B disk (user files)
*2311,2314	190**	DSK6	C disk (user files)
1403	00E	PRN1	line printer
2540	00C	RDR1	card reader
2540	00D	PCH1	card punch
*2400	180	TAP1	tape drive
*2400	181	TAP2	tape drive

* The 2311 or 2314 for the temporary disk, the A, B, and C disks, and the two 2400 tape drives are optional

devices; they are not included in the minimum configuration.

** The specified virtual addresses may be changed at any time by the CMS LOGIN command.

After entering the real device addresses, the CMS initialization message is typed

```
CMS .. VERSION n LEVEL m
```

Issue the CMS command

```
FORMAT P ALL
```

to format the P-disk to be used for CP-67 SYSGEN. The card punch and printer should be placed in READY status.

Place the CP distribution tape on symbolic drive TAP2 and issue

```
TAPE LOAD 4
```

All necessary CP files are loaded onto the P-disk. These files are of the following filetypes for generation of CP-67 and the utilities:

SYSIN files, which are the CP source decks written in Assembler Language;

EXEC files, which contain procedures for assembling and updating;

TEXT files containing the assembled modules of the system;

MACLIB files containing CP-67 macros for assembling the system and the real machine configuration;

ASP360 and COPY files for creating and changing the macro library CPMACS MACLIB, when necessary;

LOADER files containing relocating loaders with different printer addresses (that is, 000, 00E, 00F, 010, 030, 040);

SAMPLE files containing examples of the various decks required for system setup.

It is from this P-disk (on which the above CP files are loaded) that the CP system is obtained.

Note. While executing CMS on the real machine, the last card of any punched deck must be manually run out from the card punch.

SETUP OF MACHINE CONFIGURATION

The real machine configuration is described to CP via a text deck (object deck). This text deck is the result of assembling a group of real I/O definition macros with the CMS Assembler (F). A real machine configuration is contained among the P-disk files as RIOCSCTEXT. The configuration that RIOCSCTEXT supports is as follows:

	<u>Device</u>	<u>Address</u>	
	1052	009	
	1403	010	
	2540R	011	
	2540P	012	
	2703	030-03F	(8 level ASCII, 110 bps Lines *)
		040-04F	(2741 and 1050 Lines)
		050-05F	(2741 and 1050 Lines)
		060-06F	(2741 and 1050 Lines)
		070-07F	(2741 and 1050 Lines)
		080-087	(2741 and 1050 Lines)
3	2400's	0C0 - 0C2	
4	2703 BSC	088-08B	
2	2311's	390 - 391	
	2314	230 - 237	
	2314	260 - 267	
	2314	330 - 337	
	2301	101	
	2301	100	

* The customer is responsible for terminal compatibility with this program. IBM assumes no responsibility for the impact that any changes to the IBM-supplied products or programs may have on terminals provided by others.

If the above configuration does not match the installation's configuration, a real I/O source deck must be constructed by using the macros specified in Section IV and the deck must be read in and assembled. Sample simplex and duplex machine configurations are contained among the P-disk files with the identifiers of SIMPLEX SAMPLE and DUPLEX SAMPLE.

To get the real I/O definitions for the target configuration read onto the P-disk, from the card reader, place the RIO deck into the reader, ready it, and issue the

The assembled modules distributed with this system contain the following options:

&RTIMR	SETA	6
&V67	SETA	1
&ISAM	SETC	'YES'

If these options are not desired or are to be changed, update the LOCAL COPY file and reassemble the modules affected, as described for each option below.

Real Timer--&RTIMR

The real timer option &RTIMR provides for updating a virtual machines timer when the machine is in wait state, if that virtual machine has the REAL TIMER option in the DIRECTORY. This option allows a dormant machine (for example, APL) to be awakened by a timer interrupt. (See CP-67 Operator's Guide--"Directory Creation"--for details.)

The module to be assembled for the &RTIMR option is DISPATCH.

For the real timer option the SETA symbol is set to the number of concurrent real timers that the installation is willing to support. For instance, the statement

```
&RTIMR SETA 6
```

produces code in the dispatcher to maintain up to six real timers.

Virtual 360/67 Option--&V67

The virtual 67 option, &V67, provides code in the CP-67 nucleus to support virtual machines in which CP-67 can be run. Those virtual machines with the virtual 67 option specified in the directory have the facility to operate in virtual extended PSW mode with 24-bit addressing. The distributed system has the &V67 option selected. If the option is not desired, change the LOCAL COPY value to

```
&V67 SETA 0
```

and use the exec procedure CPUPASM to reassemble the following modules:

```
CFSMAIN  
DISPATCH  
IOINT
```

LOGON
MVIOEXEC
PAGTRANS
PROGINT
QUEVIO
RESINT
PSA
UNSTIO
USEROFF
VIOEXEC

ISAM Option--&ISAM

The &ISAM option allows the user to do I/O operations involving self-modifying CCW commands on DASD device such as used by OS/360 ISAM. Only those users with the ISAM option in the DIRECTORY have their CCW strings checked for self-modifying operation; thus not all users incur the additional CP overhead.

The module to be assembled for the &ISAM option is CCWTRAN. The distributed version has the option selected.

If the option is not desired, change the LOCAL COPY value to

```
&ISAM   SETC   'NO'
```

and issue the CMS command

```
CPUPASM CCWTRAN
```

Note. The above procedure for options can also be applied to code added by an installation. The module can be assembled to selectively contain the installation's code (1) by placing a COPY OPTIONS and a COPY LOCAL instruction before the START card in the module being modified, (2) by adding a SETA instruction in the LOCAL COPY file, that is,

```
&RTIMR   SETA   1
```

and (3) by checking for the setting of the option in the module.

OBTAINING CP-67

To obtain the tailored CP-67 system, a card load deck must be punched out and then loaded onto a properly formatted volume. To punch the CP card load deck, the CP utilities and card loaders, verify that the card punch is in READY status, then issue the CMS command:

where

- &1 is the address of the printer that the loader uses to print the CP load map. A loader called RELDRxxx punches at the start of the deck, where xxx, specified in &1, is the printer address in 3 hex characters (for example, 00E or 030).
- &2 is the full name of the real I/O configuration deck that was specifically assembled for the system in the procedure above (for example, RIOCS or RIOABC).

The above command punches out the complete CP load deck (about 2500 cards with each text deck uniquely identified in columns 73-75) and each deck is separated by an identifying card with the format

OFFLINE READ fname ftype

where fname is the object deck filename and ftype is TEXT.

Note. There is a pause after punching the CP-67 LOAD DECK to allow the user to clear the punch and get the last card. The user can continue by typing "ready" or he may stop here by typing "quit". This is clearly stated on the terminal as the CPGEN operation proceeds.

OBTAINING THE UTILITIES

The utilities supplied with CP-67 are as follows:

BUZZARD
DIRECT
FORMAT
SAVESYS
VDUMP

The CP utilities are punched out as part of the CPGEN procedure above. The second file output contains all the utilities, each identified by a deck separator of the OFFLINE READ form (as outlined above for the CP load deck).

The CP utilities, BUZZARD, FORMAT and SAVESYS, are stand-alone programs and must have the appropriate loader on the front that reflects the correct printer address. The

DIRECT utility runs stand-alone or under CMS. If it is run stand-alone, it requires a loader. The VDUMP utility runs only under CMS. It is used to retrieve CP-67 dumps from disk after an ABEND in the CP supervisor. The VDUMP utility is run by the user specified in the SYSGEN macro of RIOxxx as the SYSDUMP parameter. In addition to the standard CMS machine configuration, the user must also have a printer reader device in his virtual machine as

UNIT OF1,RPRT

in order to run VDUMP. (See CP-67 Operator's Guide--"VDUMP" under "Utility Operation", and "Directory Creation"--for details.)

To obtain copies of the loader, issue the CMS command

OFFLINE PUNCH name LOADER

where name is either RELDR000, RELDR010, RELDR030, RELDR00E, RELDR00F, or RELDR040, depending on the real address of the printer.

If an installation has a printer of any other address, use the RELDR000 LOADER and immediately after the loader and before the TEXT file(s) to be loaded place the following control card:

column	1	2	6	10
	CTL	WTR	ccu	

column 1 - 12, 2, 9 punch

where ccu is the desired printer address.

The CPGEN procedure punches out three loaders for the specified printer address &1 (that is, 00E) to be used with various loading functions and also three loaders with a zero address (RELDR000), which are useful when loading the CP utilities.

Note. If a loader RELDR000 is used, no load map is produced, but the loading function proceeds. A load map is very useful for the CP-67 system, but of little use when loading the utilities, thus RELDR000 can be used by an installation to load a program without using the printer.

FORMATTING OF VOLUMES

Once the CP utilities have been punched out, the direct-access volumes used by the system (for paging,

spooling, nucleus residence, and directory) must be properly formatted. This is accomplished by means of the FORMAT utility. The detailed instructions for operating FORMAT are contained in CP-67 Operator's Guide.

The system's residence volume for CP is usually formatted with the label CPDSK1, as that label is required for IPL'ing by name. If it is desired to change the volume label, the file SYSTEM SYSIN must be modified to reflect the desired system's residence volume label and SYSTEM SYSIN must be reassembled and replaced in the CP load deck.

ALLOCATION AND DIRECTORY CREATION

All 2314 and 2311 volumes to be used for CP-67 spooling and/or paging must have space allocated. The system residence volume must have space allocated for the system directory and the nucleus.

Note: 2301 and 2303 drums must not be allocated since they are specially formatted for dynamic paging allocation. The DIRECT utility (ALLOCATE control cards) is used to allocate space for the system residence volume and others, if necessary. (See CP-67 Operator's Guide for details). For 2314 residence, the following space must be allocated as permanent:

Cyl.	Usage
0	IPL and file directory
'SYSERR'	error recording (1 cyl required)
'SYSWRM'	warm start control (1 cyl required)
'SYSDNC'	nucleus residence (2 cyl required)
to 'SYSDNC+1'	nucleus residence

The values for the symbolic cylinder locations are defined in the RIOxxx deck with the SYSRES macro.

For 2311 residence, the following space must be allocated as permanent:

Cyl.	Usage
0-1	IPL and file directory
'SYSERR'	error recording (1 cyl required)
'SYSWRM'	warm start control (1 cyl required)
'SYSDNC'	nucleus residence (5 cyl required)
to 'SYSDNC+4'	nucleus residence

The symbolic values are obtained from the SYSRES macro specified in the real machine configuration description.

Cylinder 0 and others may be allocated as DRCT space. In addition, some cylinders may be allocated as permanent for user file space or named system residence. Other cylinders must be allocated as TEMP (for spooling or paging) or TDSK.

The directory must be created after the direct-access volumes are properly formatted. The detailed instructions for setting up a directory deck are contained in the CP-67 Operator's Guide. A sample directory is contained on the P-disk in the file DIRECT SAMPLE. To obtain a printout of that file, issue the CMS command

OFFLINE PRINT DIRECT SAMPLE

The directory is created by the utility DIRECT, the instructions for which are in the operator's guide. The allocation of space on the residence volume (as described above) must precede the directory creation.

CREATION OF PERMANENT RESIDENCE VOLUME

Once the volumes have been formatted and the directory created, the permanent CP residence nucleus should be created from the CP LOAD DECK. The last module in the CP load deck is the SAVECP module (serialization SCP). The function of this module is to create, after the load through the card reader, an IPL'able copy of CP on the residence volume from the core image. The volume address, label, and nucleus cylinders are indicated to SAVECP from information contained in the RIOxxx module, in particular, the information from the SYSRES macro.

To create the permanent CP residence volume, ready the CP load deck into an available reader. The first module in the load deck must be the relocating loader, which prints the load map of the system onto the printer. Perform an IPL sequence on the card reader. The volume must be mounted and ready before the load completes. It is advisable to clear core before loading. If the disk is not ready, the message

```
**** DISK CUU NOT READY ****
```

is printed.

If the label on the disk is incorrect, the message

```
**** VOLID NOT dlabel ****
```

is printed.

If either message is printed, remedy the situation and press the EXTERNAL button to retry, or start from the card load.

At the termination of the load and the creation of the residence volume, the following message appears on the operator's terminal:

DISK LOAD OK

At this time, the permanent CP residence volume has been created and may be IPL'ed to initiate system operation.

OBTAINING CP-67 FROM AN EXISTING CMS

For those installations that have an operating CP-67/CMS system, the CMS restore procedure outlined above can be bypassed. The following steps can be used to obtain CP-67:

1. Mount the distribution tape on an available tape drive attached to the desired userid as '181'.
2. Using CMS issue a TAPE LOAD 4 command. The four files will then load in their entirety onto the user's P-disk space.

Note. About 40 cylinders of 2314 space are adequate to hold all the files and do assemblies.

3. Perform any necessary assemblies for configuration or options as outlined under "Setup of Machine Configuration" and "Selection of Options".
4. Follow the procedure under "Obtaining CP-67" using the CPGEN command.
5. BE SURE TO USE THE NEW UTILITIES AND THE NEW LOADERS WITH THE NEW CP-67.

CP-67 EXEC PROCEDURES

The following CMS EXEC procedures are distributed with the CP-67 system. These procedures should be used to generate and maintain the system.

CPUPASM

CPUPASM &1 &2

CPUPASM is used to update a SYSIN file with an UPDATE file, if it exists, and to assemble the updated file or the original if no UPDATE file exists. &1 is the name of the SYSIN file to be processed. &2 can be NODECK if no TEXT file is to be produced. The original SYSIN and UPDATE files remain. The TEXT file created is for the updated version and is named &1 TEXT if no UPDATE file is applied, or .&1 TEXT if an UPDATE was applied leaving the original &1 TEXT. No LISTING file is produced.

Note that the CPGEN/CPSYS/CPUNCH EXEC functions described below use .&1 TEXT in preference to &1 TEXT.

CPMACGEN

CPMACGEN &1

CPMACGEN creates a MACLIB, named &1 MACLIB, from all COPY and ASP360 files on the P-disk. The COPY and ASP360 files should be updated using CPUPMAC before using this command.

CPMACADD

CPMACADD &1 &2

CPMACADD adds file &2 (either ASP360 or COPY) to MACLIB named &1. The MACLIB must already exist and the ASP360 or COPY file should be updated before adding the file.

CPMACREP

CPMACREP &1 &2

CPMACREP replaces the file &2 (COPY or ASP360) in the MACLIB named &1. The MACLIB must already exist and the ASP360 or COPY file should be updated before replacement.

Note. The ASP360 file may contain more than one MACRO definition. If one macro is changed the entire file (ASP360) must be added or replaced, not the macro name.

CPGEN

CPGEN &1 &2

CPGEN is used to punch a self-loading CP nucleus, the CP utilities, and CP loaders.

&1 = 000 | 010 | 030 | 040 | 00E | 00F

Select the correct relocating loader for the installation, where the three characters indicate the system's printer address.

&2 = The name of the installation's real I/O configuration deck, for example, RIOCS.

CPUPMAC

CPUPMAC &1

CPUPMAC updates a &1 MASTER file if it exists, and produces a &1 COPY or a &1 ASP360 file. If a &1 MASTER file does not exist, CPUPMAC updates an &1 COPY or &1 ASP360 file with &1 UPDATE, saving the old file as &1 MASTER and creating the updated file as &1 COPY or &1 ASP360. There must be an &1 UPDATE file on the users disk, as well as a &1 COPY or &1 ASP360 file.

Note. This command must be used on all ASP360 or COPY files which have updates, before using CPMACGEN.

Hint. Instead of regenerating CPMACS MACLIB every time an ASP360 or COPY module is changed, it is more convenient to create a new MACLIB called MACUP MACLIB. The assemble procedures automatically access MACUP before accessing CPMACS.

CPSYS

CPSYS

This EXEC procedure is used by the CPGEN EXEC function to produce the CP-67 load deck. It contains control statements and an ordered list of the nucleus components.

CPPUNCH

CPPUNCH &1 &2

This EXEC procedure is used by the CPSYS EXEC function in the CPGEN procedure. Its function is to select either the .&1 TEXT file if it exists or the &1 TEXT file. If neither exists, an error message is printed.

CPLIST

CPLIST &1

This EXEC procedure prints one or all files from a CP-67 LISTING tape generated by the CPASMGEN EXEC procedure. The first file on the listing tape must be a TAPE LOG file. The TAPE LOG file is merely a sequential list of the names of the LISTINGS that follow. For example, see TEMP EXEC. If &1 is ALL then all files (66) are printed, otherwise the tape is searched for the desired file (for example, DISPATCH) and that file (DISPATCH LISTING) is printed.

CPASMGEN

CPASMGEN &1

This EXEC procedure is the one used to completely assemble the CP-67 system producing a printed listing and a LISTING tape. An auxiliary EXEC procedure with an ordered list of all SYSIN is necessary for complete generation. For instance, if TEMP EXEC consists of the following:

```
&1 &2 ACCTON  
&1 &2 ACNTOFF  
&1 &2 ACNTIME  
etc.
```

then the following CMS procedure assembles all the modules in TEMP EXEC:

```
TEMP EXEC CPASMGEN
```

CPTGEN

CPTGEN &1 &2

This EXEC procedure is used only when running under CP-67, since it uses the XFER function. Its purpose is to create an IPL'able tape for CP-67 disk loading. The &1 is the name of the desired loader for the printer (for example, RELDROOE or RELDRO30). The &2 is the name of realio configuration module (for example, RIOCSC). This EXEC procedure uses the CPSYS EXEC procedure to produce a load deck. A tape must be

attached as 181 and a file called TAPE LOAD is read onto the user's P-disk. The reader is PURGED for this operation. The EXEC procedure then writes the TAPE LOAD file on tape.

CP-67 DECK FORMATS

This section describes the format of the CP load deck and the utilities. The name of each deck and its serialization in columns 73-76 are specified.

CP-67 SYSTEM

The following text files are contained in the CP load deck:

<u>Module</u>	<u>Serial</u>
Relocating Loader (RELDRxxx)	
SLC 000080	loader control card--with PSA
PSA	PSA
ACCTON	ACON
ACNTOFF	ACOF
ACNTIME	ACNT
CCWTRANS	CCW
CFSCOM	CCOM
CFSDBG	CDBG
CFSIPL	CIPL
CFSMAN	MAIN
CFSPRV	CPRV
CFSQRY	CQRY
CFSSET	CSET
CFSSPL	CSPL
CFSTACH	TACH
CHKCUACT	CKCU
CONSINT	CINT
CONVRT	CVT
CPFILE	CPF
CPSTACK	CPK
CPSYM	CPSM
DEDICATE	DED
DIAL	DIA
DISPATCH	DISP
DSKDUMP	DSKD
EXTEND	XTND
FREE	FREE
IOINT	IOI
IOERROR	IERR
LINK	LINK
LOGON	LOG
LOGFILES	LGFL
MRIOEXEC	MRIO
MVIOEXEC	MVIO
PACK	PACK
PAGTRANS	PAGE
PAGEGET	PGET
PLACE	PLC
PROGINT	PROG
QUEVIO	QUEV
RDCONS	RDCN

RDSCAN	RDS
RECFREE	REC
RESINT	RST
RIOxxx	RIO
SCANUNIT	SUN
SLTSIM	SLT
STCONSIO	SCN
TMPSPACE	TMP
UNSTIO	UNS
UNTRANS	UNTR
USERLKP	ULKP
USEROFF	USE
VIOEXEC	VIOX
VSERCH	VSS
WRTCONS	WCN
ICS CPEND	loader control card--with IPL TEXT
SLC 020000	loader control card--with IPL TEXT
IPL	IPL
SLC 020800	loader control card--with CPLOCS TEXT
CPLOCS CPLC	
SLC 021000	loader control card--with SYSTEM TEXT
SYSTEM	SYS
SLC 022000	loader control card--with CHKPT TEXT
CHKPT	CHKP
SLC 023000	loader control card--with CPINIT TEXT
CPINIT	CPI
SLC 025000	loader control card--with SAVECP TEXT
SAVECP	SCP
LDT SAVENUC	loader control card

CP UTILITIES

The following files are the CP utilities:

<u>Module</u>	<u>Serial</u>
BUZZARD	BUZ
DIRECT	DIR
FORMAT	FOR
SAVESYS	SSYS
VDUMP	VDMP

MACHINE CONFIGURATION DEFINITION
FOR CP-67

A description of the physical machine must be provided for CP-67. This description is contained in the text deck RIOCS. If the distributed RIOCS text deck does not meet the installation's configuration (see "Setup of Machine Configuration"), a new real I/O deck must be constructed from the macros described below, assembled, and used in place of the distributed RIOCS text deck.

The physical (real) machine is described to CP via control blocks containing information about the input/output devices, channels, and control units. These control blocks are generated by the following CP-67 macros which are described below: REALIO, SYSRES, SYSGEN, SIMPLEX, DRCH, DRCU, DRDEV, and DMXDV.

The following conventions are used throughout this description: 1. variable information is indicated in lowercase and system key-words are indicated in uppercase letters, whereas they are written in uppercase when macros are punched in cards; 2. < and > are used to bracket choices and the brackets are not part of the macro definition (for example, RDEVPNT=<0,dlabel>) would be used to indicate that RDEVPNT=0 or RDEVPNT=dlabel could be used; 3. a pair of |'s is used to indicate an optional parameter and the |'s are not a part of the macro definition (for example, SIMPLEX |n| would be used to indicate that SIMPLEX or SIMPLEX n could be used).

CP-67 MACROS

REALIO MACRO

The REALIO macro is the first card of the deck. Its purpose is to generate the prerequisite entry points for use by the system and to give a CSECT name to the deck. Its use is

```
|label| REALIO |TITLE='Listing Header'|
```

where "label" becomes the alphabetic serialization of the text deck and title appears at the top of the pages in the listing.

Note. label must be four or fewer characters.

SYSRES MACRO

The SYSRES macro is used to describe the system residence volume. Its format is

```
SYSRES SYSRES=ccu,SYSTYPE=type,SYSVOL=volid,SYSERR=aaa,  
        SYSDNC=bbb,SYSWRM=ccc
```

where ccu is the address of the disk onto which the nucleus is written; type is 2311 or 2314,; volid is the CP-67 formatted label of the volume (that is, CPDSK1); aaa is the cylinder allocated as permanent for error recording; bbb is the starting cylinder for nucleus residence; and ccc is the cylinder for warm start control.

SYSGEN MACRO

The SYSGEN macro is used to describe certain installation variables used for system operation. Its format is

```
SYSGEN SYSOPER=userid,SYSDUMP=dumpid,SYSEMRG=xxx,  
        SYSCNSL=aaa,SYSPRT=bbb,SYSPUN=ccc,SYSCORE=dddK
```

where userid is the operator's id for AUTO LOGIN; dumpid is the user's id for whom system dumps are made available from spooling files; xxx is the 270x line address for the emergency console startup; aaa is the system console address; bbb is the system line printer; ccc is the system punch, and ddd is the storage size of the real system expressed in K.

SIMPLEX MACRO

The SIMPLEX macro follows the unit, control unit, and channel specifications for any given CPU. It provides for the creation of the necessary tables required by CP and is dependent on the configuration specified. If a simplex configuration is described, the SIMPLEX macro must be followed by the END card. If a duplex configuration is being described, two SIMPLEX macros are required: the first SIMPLEX macro is followed by the I/O definitions for another CPU; the second SIMPLEX macro is followed by the END card.

```
SIMPLEX |n|
```

where 'n' is either null for a simplex system, '1' for the first half of a duplex specification, or '2' for the second half of a duplex configuration.

I/O BLOCK DEFINITION MACROS

The following macros are used to describe the physical input/output units available:

Define Channel Macro

```
|label| DRCH |CHANPNT=<0,chlabel>|,  
RCULIST=list,CHANADD=c
```

where "label" is the symbolic label of this channel; chlabel is the symbolic label of the next channel in the channel list; list is the symbolic label of the first control unit on this channel; and c is the channel address.

Define Control Unit Macro

```
|label| DRCU |RCUPNT=<0,culabel>|,  
DEVLIST=dlabel,RCUADD=c,  
|RCUPATH=<0,path>|,CUTAIL1=tail1
```

where "label" is the symbolic label of this control unit; culabel is the symbolic label of the next DRCU macro for the next control unit on this channel, if any; dlabel is the symbolic label of the first device defined for this control unit; c is the control unit address; path is the value (hexadecimal) of the logical path for this control unit (each control unit on a channel has a unique bit of the eight bits assigned to the channel and this bit or path defines which devices are accessed through this control unit); tail1 provides a connection to this control unit from the channel--use the symbolic label of the channel on which this control unit resides.

Define Device Macro

```
|label| DRDEV |RDEVPNT=<0,dlabel>|,  
RDEVCU=culabel,RDEVADD=ccu,  
TYPE=type,DECUPATH=path,
```

where "label" is the symbolic label for this device; dlabel is the symbolic label of the next device on this control unit; culabel is the label of the control unit on which this device resides; ccu is the channel, control unit, and unit address of this device; type is the device type (specified as xxxx where xxxx is 2311, 2314, 2321, 2301, 2303, 2400, 1403, 2540P, 2540R, 2701, 2702, 2703, or 2250; "path" is identical to the control units path (RCUPATH in DRCU) for this device and defines which control unit this device uses.

Define Multiplexor Device Macro

```
|label| DMXDV RDEVADD=ccu,TYPE=type,|SAD=n|,  
|RDEVPNT=<0,mlabel>|
```

where "label" is the symbolic label of this multiplexor device; ccu is the channel, control unit, and unit address of this device; type is the device type, which is one of the following: 1052, 1403, 2540RDR, 2540PCH, 2701T, 2702T, 2703T, or TT35T (where the 2701T, 2702T, 2703T implies a 1052, 2741-BCD or 2741-correspondence terminal coming into a 2701, 2702, or 2703 and the TT35T is a teletype 33 or 35); n is the SAD address of the 2701, 2702, or 2703 and has a value of 0, 1, 2, 3, or 4 (the SAD address does not indicate terminal type); For a 2701 which does not require a SAD command, specify a value of 4. For a 2703 which ignores SAD commands, a value of 4 can also be specified. For a 2702, the correct SAD value for the particular line must be specified (0, 1, 2, and 3 are valid). Your local IBM CE can supply you with the SAD numbers for each 2702 line; and mlabel is the symbolic label of the next multiplexor device in the chain. There must be one DMXDV macro defined for each 2701, 2702, or 2703 line.

Note. The ordered arrangement of the real I/O macros is required to properly generate the correct entries and tables.

The multiplexor devices must be defined before the

selector channel devices. Each selector channel must be completely defined in terms of devices, control unit, and then channel (in that order) before the next selector channel is defined. If a simplex configuration is being described, the SIMPLEX macro must be followed by the END card. If a duplex configuration is being described, two SIMPLEX macros are required: the first SIMPLEX macro is followed by the I/O definitions for another CPU; the second SIMPLEX macro is followed by the END card.

An example of a real I/O source deck for a simplex configuration is as follows:

```

RIOS      REALIO TITLE='SAMPLE SIMPLEX'
          SYSRES SYSRES=230,SYSTYPE=2314,SYSVOL=CPDSK1,SYSERR=004, *
          SYSDNC=198,SYSWRM=202
          SYSGEN SYSOPER=OPERATOR,SYSDUMP=CPSYS,SYSEMRG=020, *
          SYSCNSL=009,SYSPRT=030,SYSPUN=032,SYSCORE=768K
OPCONSOL DMXDV RDEVPNT=PRINTER1,RDEVADD=009,TYPE=1052
PRINTER1 DMXDV RDEVPNT=CARDRDR1,RDEVADD=030,TYPE=1403
CARDRDR1 DMXDV RDEVPNT=PUNCH1,RDEVADD=031,TYPE=2540RDR
PUNCH1   DMXDV RDEVPNT=TERM01,RDEVADD=032,TYPE=2540PCH
TERM01   DMXDV RDEVPNT=TERM02,RDEVADD=020,TYPE=2702T,SAD=1
TERM02   DMXDV RDEVPNT=TERM03,RDEVADD=021,TYPE=2702T,SAD=1
TERM03   DMXDV RDEVPNT=TERM04,RDEVADD=023,TYPE=TT35T,SAD=2
          .....
          .....
TERM4E   DMXDV RDEVADD=04E,TYPE=2703T,SAD=4
DRUM2    DRDEV RDEVVCU=R2820A,RDEVADD=100,TYPE=2301,DECUPTH=80
R2820A   DRCU  DEVLIST=DRUM2,RCUADD=0,CUTAIL1=CHAN1,RCUPATH=80
CHAN1    DRCH  RCULIST=R2820A,CHANADD=1,CHANPNT=CHAN2
DISK30   DRDEV RDEVVCU=R2314,RDEVADD=230,TYPE=2314,DECUPTH=40, *
          RDEVPNT=DISK31
DISK31   DRDEV RDEVVCU=R2314,RDEVADD=231,TYPE=2314,DECUPTH=40, *
          RDEVPNT=DISK32
          .....
          .....
DISK37   DRDEV RDEVVCU=R2314,RDEVADD=237,TYPE=2314,DECUPTH=40
R2314    DRCU  DEVLIST=DISK30,RCUADD=3,CUTAIL1=CHAN2,RCUPATH=40
CHAN2    DRCH  RCULIST=R2314,CHANADD=2,CHANPNT=CHAN3
DRUM1    DRDEV RDEVVCU=R2841B,RDEVADD=393,TYPE=2303,DECUPTH=80, *
          RDEVPNT=DISK4
DISK4    DRDEV RDEVVCU=R2841B,DEVADD=390,TYPE=2311,DECUPTH=80, *
          RDEVPNT=DISK5
          .....
          .....
R2841B   DRCU  DEVLIST=DRUM1,RCUADD=9,CUTAIL1=CHAN3,RCUPATH=80, *
          RCUPNT=R2250
SCOPE1   DRDEV RDEVVCU=R2250,RDEVADD=306,TYPE=2250,DECUPTH=40
R2250    DRCU  DEVLIST=SCOPE1,RCUADD=0,CUTAIL1=CHAN3,RCUPATH=40
CHAN3    DRCH  RCULIST=R2841B,CHANADD=3,CHANPNT=CHAN0
TAPE1    DRDEV RDEVVCU=R2400A,RDEVADD=0C0,TYPE=2400,DECUPTH=80 *
          RDEVPNT=TAPE2
TAPE2    DRDEV RDEVVCU=R2400A,RDEVADD=0C1,TYPE=2400,DECUPTH=80
R2400A   DRCU  DEVLIST=TAPE1,RCUADD=C,CUTAIL1=CHAN0,RCUPATH=80

```

```
CHAN0 DRCH RCULIST=R2400A,CHANADD=0,CHANPNT=CHAN0A
D2701A DRDEV RDEVCU=R2701A,RDEVADD=010,TYPE=2701,DECUPATH=80
R2701A DRCU DEVLIST=D2701A,RCUADD=1,CUTAIL1=CHAN0A,RCUPATH=80
CHAN0A DRCH RCULIST=R2701A,CHANADD=0,CHANPNT=CHAN0B
D2701B DRDEV RDEVCU=R2701B,RDEVADD=012,TYPE=2701,DECUPATH=80
R2701B DRCU DEVLIST=D2701B,RCUADD=1,CUTAIL1=CHAN0B,RCUPATH=80
CHAN0B DRCH RCULIST=R2701B,CHANADD=0
SIMPLEX
END
```

Note. * Nonshared multiplexor devices, except for tapes, should be defined as separate channel, control unit, and device blocks (macros); that is, each device has its own set, as shown for D2701A and D2701B above. This avoids lock-out problems on that channel if unusual I/O operations are performed by users on a particular device.

Further examples of REAL I/O configuration decks are available from the CP-67 distributed P-disk files: SIMPLEX SAMPLE, DUPLEX SAMPLE, and RIOCSY SYSIN .

GENERATING "NAMED" OPERATING SYSTEMS
UNDER CP-67

Providing a user with the capability of performing an IPL function by name, rather than by device, requires the installation management to save a copy of this system on a DASD device in paging form. The self-loading deck which performs this is entitled SAVESYS. It accepts a control card of the form

```
SAVE VOLID=cccccc,UNIT=ccu,FP=nn,LP=mm,CYL=ppp,DISK=tttt
```

where

cccccc is the volume label of the receiving DASD volume;

ccu is the channel, control unit, and device address of the receiving volume;

nn is the page number of the first page to be written (in hexadecimal);

mm is the page number of the last page to be written (in hexadecimal).

ppp is the starting cylinder of the image;

tttt is the device type (either 2311 or 2314).

The space indicated on the card must have been previously allocated as permanent space on the volume (see "SAVESYS" under "Utility Operation" in the CP-67 Operator's Guide, for format conventions). The information on the control card must match that information provided in the SYSTEM module assembled and loaded with the system. In that module the system name, location, shared pages, if any, and operating conventions are established. In the distributed SYSTEM module, the system name is CMS, and eighteen pages are saved, beginning at page 0, up to and including page X'11'. The saved copy of CMS is written on the 2314 volume labeled CPDSK1 at cylinder 200.

The SYSTEM SYSIN file is constructed using a SYSTEM macro, which has the format

```
SYSTEM NAME=nnn,FIRSTP=aa,LASTP=bb,TABLE=ttt,  
VOLID=xxx,EXADD=ccc,SYSMASK=sss,RUNKEY=kkk,  
SYSTAB=yyy
```

where

nnn is the name for the system;

aa is the first page saved;
 bb is the last page saved;
 ttt is the label of the SWPTABLE macro associated with this system;
 xxx is the label of the volume on which the system is saved;
 ccc is the virtual machine execution address;
 sss is the virtual machine system mask;
 kkk is the virtual machine protection key;
 yyy is the label of the shared page table in PAGTRANS if this system has shared pages; otherwise it is zero.

The SWPTABLE macro used with the SYSTEM macro has the following format:

```

ttt SWPTABLE DASDORG=ccc,FIRSTP=aa, LASTP=bb,
      FIRSTSP=xx, LASTSP=yy, DISK=ddd
  
```

where

ttt is a label (same as ttt in SYSTEM macro);
 ccc is the cylinder address where the system is saved;
 aa is the first page saved;
 bb is the last page saved;
 xx is the first shared page, if any;
 yy is the last shared page, if any;
 ddd is the disk type where saved, 2311 or 2314.

The SYSTEM macro builds a table to define the name and running attributes of the saved system. The SWPTABLE macro builds a model SWPTABLE that is mapped to the virtual machine SWPTABLE in core to give that machine access (through paging) to the saved system.

The number of cylinders required to hold a named system depends upon the number of pages saved and the device type. For a 2314, one cylinder holds 30 pages; a 2311 cylinder holds 8 pages. Thus, CMS with 17 pages requires one 2314 cylinder or three 2311 cylinders.

When SAVESYS saves a copy of the system, it saves from FP to LP. If LP is greater than X'20', the SAVESYS module must be loaded into higher core, as it normally loads into X'20000'. To load SAVESYS into higher core, change the address in the SLC 20000 card at the front of the SAVESYS text deck.

The procedure for creating the page-form copy is as follows:

1. Load the required system into memory, with the address stop switches set to a location within that system where execution may be resumed without the system

assuming any previous state in the machine (that is, registers, lower core locations, etc.). For CMS the address stop location is X'F0'.

2. When the system has entered manual state (that is, the address has been reached), load the SAVESYS deck into the system card reader and load from the reader.
3. If the save was successful, a message appears on the operator's terminal to that effect.

Note. The following procedure must be used to run Version 2 of CMS under Version 3 of CP as the named system CMS: either the SYSTEM module must be changed in CP, as Version 2 of CMS has only 3 shared pages, or the saved copy of Version 2 of CMS must be given a name other than CMS.

See "Saving CMS under CP-67" in this manual for details of preparing CMS for the CP-67 SAVESYS function.

CP-67 MAINTENANCE PROCEDURE

When corrections have to be made to a CP-67 module, a PTF is made available to the installation through the normal channels. The PTF is in a form suitable for a CMS OFFLINE READ or TAPE LOAD function depending upon the distribution medium. A special EXEC procedure is distributed to be used when applying PTF's. A description of the use and function of this procedure is given below.

CPPTF EXEC

CPPTF &1 &2

where &1 is the name of the CP-67 module to receive the PTF and &2 is the PTF number (for example, CPPTF DISPATCH A32486CA). Having first loaded the PTF on to the maintenance P-disk (using OFFLINE READ or TAPE LOAD) the user then issues all necessary CPPTF EXEC procedures to effect the change. One PTF may involve changes to several modules. For instance, PTF number A32486CA may create (through loading on to the P-disk) the following files:

DISPATCH	A32486CA
PAGTRANS	A32486CA
UNSTIO	A32486CA

The user must then issue the following CMS commands:

CPPTF DISPATCH A32486CA
CPPTF PAGTRANS A32486CA
CPPTF UNSTIO A32486CA

The function of the CPPTF is as follows:

The PTF file (for example, DISPATCH A32486CA) is applied to the corresponding SYSIN file (that is, DISPATCH SYSIN), using the CMS UPDATE program. The PTF creates a new SYSIN file replacing the original. If the installation wishes to preserve the original SYSIN file, it should be saved beforehand or copied and given a new name. Once the new SYSIN has been created, it should remain on the disk (or be available) since further PTF's that may be applied to the module may require that a previous PTF be applied first. The CPUPASM EXEC procedure is then invoked to assemble the new SYSIN file. If the installation has its own UPDATE file, CPUPASM applies it to the now modified SYSIN. The installation should check that any UPDATE file interfaces correctly to the SYSIN file once PTF's have been applied. The CPPTF EXEC procedure finally punches the &1 TEXT file, prints a completion message, and exits.

TUNING OF CP-67

To prevent paging overload, CP-67 allows only a subset of the users to be dispatched at one time. There are two queues for such users; both queues are served round-robin, and the queued user remains in the queue until he has used an allotted amount of computer time. A user enters QUEUE1 whenever he issues a carriage return from his terminal. He stays in QUEUE1 until the time quantum (400 milliseconds) is used or his virtual machine requires more console activity. If a user exceeds the QUEUE1 time quantum, he is placed in QUEUE2 or queued to be placed in QUEUE2, which has a time quantum of five seconds. The QUEUE1 time quantum is set to allow an average edit request to be serviced.

The paging algorithm is tied to this dispatching system. Whenever a page of memory is required, PAGTRANS uses a page assigned to a user who is not in QUEUE1 or QUEUE2 (and thereby paging runnable against unrunnable users). The intent of this queue structure is to set a limit on the number of tasks that can compete for computer resources (memory and cpu), and to segregate short execution tasks (for example, edit requests) from long execution tasks (for example, FORTRAN compilation). By so segregating the tasks, limiting the number of long execution tasks, and minimizing paging between runnable users, excessive paging

can be avoided.

The queue sizes are automatically adjusted at system IPL time based upon real core size. The queue sizes for various core capacities are

	256K	512K	768K	1M
Q1	3	6	9	12
Q2	1	3	6	9

The operator has a console function to change the size of Q2.

INSTALLING CMS

DUMP/RESTORE UTILITY

To create a CMS System disk, place the distribution tape on a tape drive, address 0CUU. Clear core and IPL the tape by pressing the LOAD button. When the wait light remains on, and the system and load lights go off, hit REQUEST on the online console. DUMP/RESTORE is given control.

CMS may be restored under CP-67, if CP already exists. Merely, attach the tape to a virtual machine and issue IPL ccu. Produce an ATTENTION interrupt.

The console address is dynamically set. If replies are incorrect while the DUMP/RESTORE is executing, DMPRST reinitializes itself and repeats Question 1. If an I/O error is sensed on the disk or tape, it tries to rectify the error condition. If successful, it continues. If not, it prints out a message and terminates.

Termination, either after a successful operation or because of an error, places the CPU in the wait state.

Q1: TASK? ("DUMP" , "REST"):

DUMP--disk to tape
RESTORE--tape to disk

Q2: DEVICE TYPE? ("2311" , "2314"):

specify the disk device type

Q3: TAPE ADDRESS? ("0CUU"):

self-explanatory. Must enter a four hexadecimal reply (for example, 0181, 0275)

Q4: DISK ADDRESS? ("0CUU"):

self-explanatory. The reply must be four hexadecimal digits (for example, 0191,0290).

Q5: REWIND TAPE? ("YES" , "NO"):

This applies to before and after the operation. Obviously, since the tape after IPL is positioned at its second file, the answer must be NO.

- Q6: NUMBER OF CYLINDERS? ("ONNN"):
 self-explanatory. The reply must be four decimal
 digits (for example, 0203, 0100). For restoring the
 distribution tape, specify 203 cylinders for a 2311
 or 54 cylinders on a 2314.
- Q7: STARTING CYLINDER? ("ONNN"):
 self-explanatory. (For the system disk, reply:
 "0000".)
- Q8: BEGINNING OF TAPE? ("YES", "NO"):
 The answer determines if a tape mark is to be written
 ahead of the file. If YES, a tape mark is written; if
 NO, the EOF at the end of the previous file serves as
 the tape mark for the beginning of this file. (When
 restoring the distribution tape, Q8 is not asked.)

The DUMP/RESTORE operation begins after the last
 question is answered. If either device is not ready, the
 program waits for its ready state. When restoring, the disk
 need not be formatted or DASDI'ed. DMPRST formats the disk
 as it is being restored.

Upon encountering an EOF, satisfying the amount of
 cylinders specified in Q6, or executing a SEEK beyond the
 disk limits, the operation terminates with the message

'DUMP/RESTORE MOVED nnn CYLINDERS
 THERE WERE nnn RECOVERABLE TAPE ERRORS.'

```

-----
| TASK? ("DUMP","REST"): restore |
| DEVICE TYPE? ("2311","2314"): 2314 |
| TAPE ADDRESS? ("0CUU"): 0181 |
| DISK ADDRESS? ("0CUU"): 0190 |
| REWIND TAPE? ("YES","NO"): no |
| NUMBER OF CYLINDERS? ("ONNN"): 0054 |
| STARTING CYLINDER? ("ONNN"): 0000 |
| BEGINNING OF TAPE? ("YES","NO"): no |
| DUMP/RESTORE MOVED 054 CYLINDERS. |
| THERE WERE 001 RECOVERABLE TAPE ERRORS. |
-----

```

Figure 1. Restore procedure for CMS distribution tape

CMS SYSTEM DISK

Once restored, the CMS System disk assumes the following appearance:

Flock Information

0-2 IPL pointers, CCW's, etc.

3 label = CMS190

4 Master File Directory

5-7979

1. Nucleus Text decks in one file called NUCLEUS 3.0, in which each individual text deck is preceded by an OFFLINE READ filename TEXT card. The nucleus text decks are also present individually.
2. Disk Command Modules.
3. Disk Command Texts.
4. EXEC Procedures

7980-8100

IPL'able Nucleus (0-12000x and loader tables).

The nucleus routine, NUCON, contains a device table. It has been initialized as follows:

S-Disk	=	0190
P-Disk	=	0191
T-Disk	=	0192
CONSOLE	=	0009
CREADR	=	000C
PRINTER	=	000E
CPUNCH	=	000D
TAP1	=	0180
TAP2	=	0181
DSK4	=	0000
DSK5	=	0000
DSK6	=	019C
CRT1	=	0106

These values are the standard CMS machine device addresses.

= SYSIN) of every CMS routine.

ALTERING THE CMS SYSTEM DISK

NUCLEUS

After restoring the CMS Source Disk, IPL CMS. Use the CMS Source Disk as a P-disk, 191. Log in and issue the command

OFFLINE PUNCH NUCLEUS 3.0.

The file punched, about 1800 cards, contains all the text decks that constitute the CMS nucleus. The cards should now be interpreted. Separating the individual text decks is an "OFFLINE READ name TEXT" card. Keep this nucleus deck. It is needed whenever changes are to be made to the contents of the System disk. (NUCLEUS 3.0 is also on the CMS System disk.)

The LOADER deck supplied is used to IPL the nucleus under CP (that is, it recognizes the standard CMS virtual device addresses). If the nucleus is to be IPL'ed on incompatible device addresses, the loader deck (first deck) must be replaced with the appropriately addressed loader.

The CMS nucleus may also be created by using the two EXEC procedures on the system disk--JPUNNUC and NUCLEUS3. Issue the command NUCLEUS3 EXEC JPUNNUC userid/OFF. Each component text deck of the nucleus is punched. If a userid was specified, the deck is XFER'ed to that userid. If OFF was specified, the physical card deck is punched.

IPL'ABLE SYSTEM DISK

To rewrite the IPL'able nucleus onto the CMS System disk, place the nucleus in the card reader and press the LOAD button. Under CP, read the nucleus deck into the CP card reader with the appropriate ID card, or use the EXEC procedure NUCLEUS3 and XFER the deck into the card reader, then type the CP command IPL C. The nucleus is loaded into core, linkages and V-cons are resolved, and a load map is produced. An IPL'able core-image copy of the nucleus (locations 0-12000) may now be written onto disk.

After the IPL sequence has completed, the IPLDISK routine is given control. It expects the standard console (terminal) address (009). If it is other than 009, the wait state is entered, and IPLDISK accepts an ATTENTION interrupt in order to define the console address. The following information is then sought:

"REWRITE NUCLEUS? (YES,NO)":

YES--a new IPL'able copy of the CMS nucleus is to be written onto disk

NO --no writing takes place. Control returns to INIT (CMS initialization routine)

"DEVICE ADDRESS? (OCUU,SYS)":

A core-image copy is written onto device OCUU (if specified) or onto SYS, which is the System disk address specified in the nucleus device table.

"STARTING CYLINDER? (0XXX,SYS)":

The core-image copy is placed at cylinder 0XXX--hexadecimal representation--or on the SYS cylinder. The SYS cylinder is

	<u>cc</u>	<u>hh</u>	<u>rr</u>	<u>block</u>
2314:	53	04	00	7980
2311:	199	05	00	7980

"VERSION IDENTIFICATION (15 bytes)":

Fifteen bytes of information, including blanks, are used as the version title and are printed out when IPL'ing the CMS nucleus.

An IPL'able copy of the CMS nucleus (locations 0-12000) is written onto the specified disk. A load map showing all deck names and reference points for the nucleus is written to the system printer device. There are four unresolved symbols at the end of the load map, as follows:

LOGIN, OFFLINE--may be either nucleus or disk-resident commands.

BATCH, SYSCTL--external references for the Batch Nucleus.

SAVING CMS UNDER CP

In place of IPL'ing a device each time CMS is to be invoked, CP allows the system programmer to SAVE a copy of

the CMS nucleus, so that a user may invoke CMS by issuing the CP command IPL CMS.

To save CMS, IPL the CMS System disk on the bare machine (that is, not under CP). Cause an ATTENTION interrupt on the console.

The following is typed:

- Q1: REDEFINE DEVICE ADDRESSES? ("YES", "NO")... (reply NO). This question is asked when running CMS on the bare machine to allow the user to dynamically change the standard device addresses.
- Q2: WILL THE CP-SAVE FUNCTION FOR CMS BE EXECUTED? ("YES", "NO")... (reply YES).
- C1: ALL DEVICE ADDRESSES MUST BE ENTERED AS 4 HEXADECIMAL DIGITS: "0Cuu".
- Q3: ENTER THE "REAL" address of the CMS SYSTEM DISK... reply "0CUU", where CUU is the real online address of the CMS system disk--as opposed to virtual address 190.
- Q4: SET THE "ADDRESS COMPARE" SWITCH FOR LOC X'F0'; REPLY "GO" After setting the instruction counter switches to reflect hexadecimal location 'F0', and setting the "address compare" key "on", reply "GO".
- C2: WHEN "MANUAL" STATE IS ENTERED, IPL THE CP "SAVESYS" UTILITY DECK FROM THE CARD READER

When the CPU stops at location X'F0', place the CP SAVESYS utility deck with loader and control card into the real card reader, reset the load address on the CPU console, and IPL the card reader. The message NORMAL TERMINATION OF SAVE FUNCTION is typed.

CMS EXEC PROCEDURES

-GEND is used to create overlay structures for the OS language processors, using the object decks and producing modules. These -GEND EXEC procedures

are used only when creating the overlay module structure. Do not try to use them when the text decks of the processor components are not present. For re-genmoding the interface modules, use the respective GENDALL EXEC segment. No arguments need be passed to -GEND procedures.

GENDALL

indicates which modules contain auxiliary directories. When changing the base command module, or moving the secondary processor modules, the GENDIRT command must be used. GENDIRT is used to complete auxiliary file directories. It is to be used by the system programmer when creating modules containing auxiliary directories. GENDALL requires the code name of the specific language processor to be passed as the only parameter (for example, ASSEM, FORT, PLI).

CMSEGEN

shows the method of creating all CMS disk-resident modules. The name of the disk module to be created must be passed to CMSEGEN (for example, to create the TAPE module, issue the command CMSEGEN TAPE).

NUCLEUS3/JPUNNUC

is used to create a physical CMS nucleus deck by punching each individual TEXT deck. Issue the command: NUCLEUS3 EXEC JPUNNUC userid, where userid is the identity of the virtual machine to which CMS nucleus deck is XFER'ed.

MAINTAINING THE CMS SYSTEM DISK

To add or change commands in the nucleus or on the system disk, proceed as described below.

CMS MAINTENANCE PROCEDURES

To update the Version 3 CMS system the procedure described below must be applied.

When a PTF is received, the user should apply it to his Version 3 SYSIN deck. The PTF is an update deck identified as "filename AxxxxyCA", where filename is the name of the routine to which the update is to be applied and AxxxxyCA is the filetype. xxxx is the APAR number, and y is the APAR code that the PTF corrects.

Example:

```
V3 deck      'edit sysin'  
Update      'edit AxxxxyCA'
```

The command to issue is UPDATE EDIT SYSIN EDIT AxxxxyCA (seq8 inc P)

The "seq8" allows sequencing of up to eight digits. The "inc" includes the sequence number of the update card into the new updated sysin file. The P causes a newly updated SYSIN file to be created.

The updated file created is "edit sysin".

Note. The original "edit sysin" is replaced with the updated copy.

CMSPTF EXEC :

The updating process described previously can be executed with the aid of a CMS EXEC file called CMSPTF EXEC.

This EXEC procedure takes the original SOURCE deck and the PTF deck and creates the new SOURCE deck, assembles it producing a TEXT deck, prints an UPDATE LOG of the update changes, and an assembly LISTING onto the offline printer, and offline punches the TEXT deck.

This enables the user to update and get a copy of the new object deck in a one-step procedure.

CHANGING DISK-RESIDENT COMMANDS

As above, obtain the object deck of the new command.

Using the system disk as a P-disk, offline read the text deck of the new command. Then create a workable module by issuing

```
LOAD name (TYPE CLEAR)
GENMOD name (P2)
```

All system modules must have a filemode of P2.

If the new command is a transient routine, issue the following:

```
ERASE name MODULE
LOAD name (TYPE TRANS)
GENMOD name (P2)
```

All system modules must have a filemode of P2.

The CMSGEND EXEC procedure may be used to generate any CMS disk-resident command. Merely issue the command "CMSGEND name", where "name" is the title of the TEXT and MODULE of the desired routine.

C-DISK AS A READ-ONLY EXTENSION OF S-DISK

In some installations, it may be desirable to have an extra disk as a read-only extension of the standard S-Disk. This may prove useful for one of several reasons:

1. If the system customarily has a large number of users, it can provide a more efficient utilization of system resources to place some of the system disk programs and libraries on one disk pack, and others on another pack. If some of the programs and/or libraries currently on the S-Disk are placed on another disk instead, this can be accomplished.
2. If a system disk gets very full from addition of new programs, it may be advisable to put new added programs on an extra disk instead of adding more cylinders to the S-Disk.
3. It may be that certain new or restricted programs are to be made available to a limited number of users. In this case, such programs could be put on the extra disk, which would only be in the directory of those permitted to use such programs.

CMS now has the capability to provide such an extra disk as a read-only extension of the S-Disk. This extra disk is the C-Disk, and has a default disk-address (in the NUCON table) of 019C. If it is desired to use the C-Disk as a read-only extension of the S-Disk, the following steps must be taken, and the CMS initialization procedure does the rest:

1. Compilers and any other programs which require the GENDIRT procedure must be on the S-Disk. IBM-supplied programs in CMS requiring GENDIRT currently are the Assembler, FORTRAN, and PL/I.
2. Create the desired modules on the extra disk using the same procedures you would for any read-write disk (for example, by OFFLINE READ of text decks, COMBINE, TAPE LOAD, DISK LOAD) but taking care to ensure that the modules or libraries which are to be user-accessible have a P2 mode (as on the S-Disk). (See also "Altering the CMS System Disk"--CMS EXEC Procedures.)
3. Run an update deck called CDISK UPDATE against the IADT SYSIN deck in the CMS nucleus. That is, with IADT SYSIN and CDISK UPDATE available, execute "UPDATE IADT SYSIN CDISK UPDATE (SEQ8)", assemble the resulting .IADT SYSIN, and then place the .IADT TEXT in the CMS nucleus using the procedures described above under "Changing Nucleus Resident Commands".

The aforementioned CDISK UPDATE deck changes the description of the C-Disk at ADTC in the IADT Initial Active Disk table to flag the C-Disk as a read-only extension of the S-disk, and should be as follows:

```
// R 20000
          C-DISK (READ-ONLY EXTENSION OF S-DISK)
// R 239000 240000
          DC 1C'S'          C-DISK = READ-ONLY EXTENSION OF S-DISK
          DC AL1(ADTFRO)    FIRST FLAG-BYTE (READ-ONLY BIT SET)
```

4. Place the C-Disk as a read-only disk in the virtual machine descriptor in the CP directory, with virtual address 19C, for all users who are to have the C-Disk available to them.

If these steps are followed, and the C-Disk is attached and ready at CMS Initialization time, while the version identification message is being typed, CMS automatically logs in the P2 files from the C-Disk, as a read-only extension of the S-Disk, in a manner similar to the SYSGEN function for obtaining the SSTAT table from the S-Disk.

The C-Disk is then a read-only extension of the S-Disk, transparent to most CMS commands. STAT C gives the

disk-statistics on the C-Disk (as STAT S does the standard S-Disk), and LISTF filename filetype C lists the C-Disk files just as LISTF filename filetype S does the standard S-Disk files. (LISTF and STAT purposely do not attempt to handle read-only extensions of disks, to avoid ambiguity.)

CMS CONSIDERATIONS

FORMATTING THE P-DISK

Once the CMS System disk is restored and all device addresses are assured correct in matching either the bare machine configuration or the CP allocation directory, issue an IPL to the System disk.

Initially, each user must issue as his first command FORMAT P ALL. This command formats his P-Disk for CMS usage. If the disk is not formatted, I/O errors occur when accessing the P-disk.

Note. All disks that are to be accessed by users must be formatted in the same manner as above.

Version 3--Version 2 Compatibility

The disk formats for Version 2 and Version 3 are compatible; however, the internal core tables are incompatible between these two versions of CMS.

Do not use old commands such as V2L0 or V2L1 which manipulate files with Version 3 (such as LOGIN, LISTF, OFFLINE, ERASE, TAPE, DISK, FORMAT, and START) as they do not work.

If a previous version of CMS is to be examined, log it in as a C-disk so that it will be searched last and its routines will not be used in place of the system routines.

OBTAINING A NUCLEUS LOAD MAP

A load map of the nucleus, giving the name and core location of each nucleus routine, may be obtained by

IPL 190 (CMS system disk)

login on any formatted P-disk

MAPPRT B OFF (see other options for this
command in CP-67/CMS User's Guide)

The file CMS-NUC ALPHANUM P1 is created on the P-disk and is also printed offline.

P-DISK CONSIDERATIONS FOR THE BARE MACHINE

When operating on the bare machine, the P-disk is used from real cylinder 0. Therefore, if a user wishes to use his own files on the bare machine, his files must be physically located on real cylinder 0.

The user must also be aware of his P-disk size when he is operating on the bare machine. If more files are created than he has disk space, CMS is not aware that the user has less than 203 cylinders and allows the user to create the files over the entire disk pack if the space is needed.

COMMAND ABBREVIATIONS

When a CMS command is issued from the terminal or from a routine, several tables are searched for the command name. If no command is defined for the name, CMS assumes that a command abbreviation was used. The nucleus-resident routine ABBREV is then searched to verify and find the full-name equivalent of the command. Command abbreviation is defined as the minimum number of characters necessary to recognize the full command name.

If standard system abbreviations are not to be allowed in CMS, the ABBREV TEXT deck should be removed from the nucleus deck.

To change, add, or delete an abbreviation, the ABBREV routine must be reassembled, altering the abbreviation table as desired. The table is constructed using the macro ABRV as follows:

ABRV name, number

where name is the full name of the command, and number is the minimum number of characters which are accepted as an abbreviation.

The macro must be used for each desired command abbreviation. For example:

```
ABRV OFFLINE, 1
```

signifies that the acceptable abbreviation for the command OFFLINE is O (also, OF, OFF, OFFL, ...).

```
ABRV ALTER, 2
```

states that the abbreviation for ALTER is AL (also, ALT, ALTE, ALTER).

INSTALLATION OF THE CMS BATCH MONITOR

From the CMS System disk obtain a copy of the Batch Monitor Nucleus, BATNUC 3.0, by issuing the command OFFLINE PUNCH BATNUC 3.0.

Set up a disk area that is used only by the virtual batch machine. This disk area, address ccu, contains the IPL'able copy of Batch nucleus and need be only two-2314 cylinders, or four-2311 cylinders. Also, set up a P-disk, 191, for the Batch machine.

Write an IPL'able copy of the Batch nucleus, BATNUC, onto disk ccu in the same manner as the CMS nucleus is reloaded onto the System disk. Two comments: 1) in reference to Q2 about device address, reply 0ccu, where ccu is the disk address specifically defined for the Batch machine; 2) for Q3, answer 0NNN, where NNN is an integer whose absolute value is greater than zero but less than the integer value of the number of cylinders allotted to the disk area.

After writing the Batch nucleus onto ccu, the message READY is typed. The Batch Nucleus disk, ccu, must be IPL'ed before executing any Batch Job Streams.

To run a CMS Batch job stream, first it is necessary to IPL CUU. The Batch nucleus may also be SAVE'd by CP, using the name BATCH. If the name BATCH is undesirable, a different name may be used by changing the CP routine SYSTEM, which contains the names and descriptions of all IPL'able named systems. Batch nucleus has set a time limit per job (five minutes) and an output limit per job (5000 pointer lines). These limits may be modified by reassembling the routine BATJCB and placing the new copy into the BATCH nucleus.

SAMPLE PROBLEM

A sample problem is distributed as File 5 on the tape containing the basic program material. It exists in TAPE DUMP form.

To load the sample problem onto the P-disk, position the tape after the fifth tape mark and load it from tape. This is done by issuing

```
TAPE REWIND
TAPE SKIP 4
TAPE LOAD
```

Then the program is ready to be compiled and executed.

The complete procedure for logging in, IPL'ing 190 for CMS, FORMAT'ing the P-disk, loading the FORTRAN source deck on the P-disk, and then compiling and executing the sample problem is shown in Figure 2. System responses are printed in uppercase; user-entered information is typed in lowercase. Figure 3 contains a listing of the file SAMPLE FORTRAN.

```

C      SAMPLE PROBLEM FOR TESTING CP-67/CMS TYPE III DISTRIBUTION
C
1      PRINT 100
100    FORMAT ('      PLEASE TYPE IN THE NUMBER OF VALUES TO AVERAGE')
      READ 101, N
101    FORMAT (12)
      IN (N) 2,99,2
2      PRINT 105
105    FORMAT ('      FIRST NUMBER? (FORMAT IS F10.4)')
      READ 103, SUM
      M=N-1
      DO 3 I=1,M
      PRINT 102
102    FORMAT ('      NEXT NUMBER?')
      READ 103, X
103    FORMAT (F10.4)
      SUM = SUM + X
3      CONTINUE
      Y = N
      AVG = SUM/Y
      PRINT 104, AVG
104    FORMAT (15X, 'AVERAGE = ',F10.4)
      GO TO 1
99     PRINT 106
106    FORMAT ('      *** TEST COMPLETE *** WELCOME TO CP-67/CMS! ')
      STOP
      END

```

Figure 3. Listing of SAMPLE FORTRAN

INDEX

Abbreviations.....43
ACCTON.....15,17
ACNTOFF.....15,17
Allocation.....10,11,42
Auxiliary directories.....37
Batch monitor.....44
BUZZARD utility.....8,18
Changing disk resident commands (CMS)..
 40,41
CMS source disk.....33,34,38,45
CMS system disk.....2,33,34,38,39,44,45
CMSPTF.....38,39
Command abbreviations.....43
Configuration.....1,2-4,5,8,9,10,12
 14,15,19,20,23,24,42
CP load deck.....7,8,10,11,14,15,17
CP residence volume.....1,10,11,12,20
CPASMGEN.....15
CPGEN.....8,9,12-15
CPLIST.....15
CPMACADD.....5,13
CPMACGEN.....13,14
CPMACREP.....5,13
CPPTF.....27,28
CPPUNCH.....13,15
CPSYS.....13-15,23
CPUPASM.....5-7,13,28
CPUPMAC.....13,14
Device address(es), CMS.....2,3
DIRECT utility.....8,18
Directory.....1,6,7,9,10,11,32,40-42
Directory creation.....6,9-11
Dispatching.....28
DMPRST.....30,31
DMXDV.....22
DRCH.....21
DRCU.....21
DRDEV.....22
Drums.....1,10
DUMP.....2,30,31,33,45
DUMP/RESTORE.....2,30,31
Duplex.....4,20,23,24
Exec procedures, CMS.....1,32-34,36-41
Exec procedures, CP-67.....1,3,6,13,27
FORMAT utility.....8,18
Formatting P-disk.....9,42
Formatting volumes.....9,10
FORTRAN.....28,41,45-47
FUNCTAB.....39
GEND.....36,37
GENDALL.....37
GENDIRT.....37,41
IADT.....41

IPL.....2,10,11,18,25,29,30,32,34-36
 42,44,46
 ISAM.....5,6,7
 JPUNNUC.....34,37
 Label.....10,11,19-22,25,26,32
 Language processor(s).....36,37
 Libraries.....40,41
 Loader....3,8,9,11,14,15,17,18,32,34,36
 LOGIN.....3,20,35,42,43,46
 Machine configuration, definition.....
 1,19
 Machine configuration, distributed.....
 19,24
 MACLIB.....3,5,13,14
 MACUP.....5,14
 Maintenance procedures, CMS.....38
 Maintenance procedures, CP-67.....27
 Message.....3,11,12,15,27,28,30,31,33
 36,41,44
 Named, system(s).....11,25-27,44
 Nucleus.....1,6,10,11,14,20,32,34-39
 41-44
 NUCLEUS3.....34,37
 NUCON.....32,41
 Offline.....5,8,9,11,27,32,34,35
 38-44
 Online.....30,36
 Operator's console.....20,33
 Options, CMS.....1,39,43
 Options, CP-67.....5-7,12
 OS.....7,36
 Pages.....19,25-27
 Paging.....9,10,11,25,26,28
 Password.....46
 Printer.....2,3,8,9,11,14,15,20,32,35
 38,39
 Punch.....1,2-4,7-9,14,20,34,39,44
 Queue1.....28
 Queue2.....28
 Reader.....2,4,9,11,16,27,34,36
 REALIO.....1,15,19,23
 Redefining device address(es).....33
 RELDRxxx.....8,17
 Restoring.....31,33,34
 RPRT.....9
 RTIMR.....6,7
 SAVECP.....11,18
 Saved system.....25-28,44
 SAVESYS.....8,18,25-27,36
 Saving, CMS.....27,35
 Serialization.....11,17,19
 Setup.....3,4,12,19
 Shared, page(s).....25-27
 SIMPLEX, macro.....20,4,19,20,23,24
 SLT.....18
 Spooling.....10,11,20

SSTAT.....41
 Startup.....20
 SYSCNSL.....20,23
 SYSCORE.....20,23
 SYSCTL.....35
 SYSDNC.....10,20,23
 SYSDUMP.....9,20,23
 SYSEMRG.....20,23
 SYSERR.....10,20,23
 Sysgen, CMS.....2,3,30,41
 Sysgen, CP-67.....2,3,9,19,20
 SYSGEN macro.....20
 SYSMASK.....25
 SYSOPER.....20,23
 SYSPRT.....20,23
 SYSPUN.....20,23
 SYSRES.....1,10,11,19,20,23
 SYSTAB.....25
 SYSTEM,module.....25
 SYSTYPE.....20,23
 SYSVOL.....20,23
 SYSWRM.....10,20,23
 Tape.....1,2,3,12,15,16,27,30,31,33
 37,41,42,45,46
 TDSK.....11
 Terminal.....4,8,12,22,27,28,33,35,39
 43,46
 Timer.....5,6
 TMPSPACE.....18
 Tuning.....28
 Userid.....12,20,34,37
 Utilities, CMS.....1,9
 Utilities, CP-67....1,3,7-9,12,14,17,18
 VDUMP.....8,9,18
 Virtual,I/O.....2,3
 Virtual 67 (V67).....5,6
 Virtual machine.....1,6,9,26,28,30,37
 41,44
 Volume label.....10,11,20,25,26

International Business Machines Corporation
 Data Processing Division
 112 East Post Road, White Plains, New York 10601
 (USA only)

IBM World Trade Corporation
 821 United Nations Plaza, New York, New York 10017
 (International)